积分方程求解及其机械化

New Algorithms for Solving Integral Equations With Mechanization

王玮明 著

甘肃科学技术出版社

社名题词　李政道

责任编辑　韩　波

封面设计　宋　烨



ISBN 978-7-5424-1234-8

定价：35.00元

积分方程求解

及其机械化

王玮明 著

兰州

甘肃科学技术出版社

图书在版编目（CIP）数据

积分方程求解及其机械化/王玮明著. —兰州：甘肃科学技术出版社，2008.11
ISBN 978-7-5424-1234-8

Ⅰ.积… Ⅱ.王… Ⅲ.积分方程—机械化—算法 Ⅳ.O175.5

中国版本图书馆CIP数据核字（2008）第163936号

责任编辑 韩 波(0931-8773237)
封面设计 宋 烨
出版发行 甘肃科学技术出版社(兰州市南滨河东路520号 0931-8773237)
印 刷 甘肃地质印刷厂
开 本 787mm×1092mm 1/16
印 张 9.5
字 数 190千
插 页 5
版 次 2008年12月第1版 2008年12月第1次印刷
印 数 1～500
书 号 ISBN 978-7-5424-1234-8
定 价 35.00元

前 言

积分方程是继微分方程之后出现的一个新的近代数学的重要分支, 也是科学研究和解决工程技术问题的重要工具之一, 具有广泛的应用. 方程求解是积分方程研究的热点和难点之一. 本书在前人研究的基础上, 利用吴文俊院士所倡导的 “数学机械化” 的思想和方法研究了积分方程求解以及机械化算法等问题. 主要工作包括如下几个部分:

第一章, 简要综述了积分方程研究简史、积分方程求解方法、数学机械化及其应用等; 第二章, 研究了 Fredholm 积分方程和方程组的豫解核求解法及其机械化算法, 利用这些算法求解 Fredholm 积分方程时, 所要做的全部事情就是输入描述方程的信息, 然后机械化算法将给出所求方程的解析解. 特别地, 在机械化求解的算法设计中, 为了增加求解过程的逻辑性和理解度, 设计了可读性求解过程输出, 使得机械化求解结果与我们用纸和笔求解方程时的过程几乎是一样的; 第三章, 研究了 Volterra 积分方程的 Neumann 级数与 Taylor 级数求解法, 以及求解 Volterra 积分方程组的迭代法. 特别是在利用 Neumann 级数法求解 Volterra 积分方程的过程中, 将得到的有限迭代核序列 $k^N(s,t)\,(N=1,2,\cdots,10)$ “分解” 为若干个部分, 对每一部分逐一运用数学归纳法, 最后再按照原来 “分解” 的逻辑顺序合并在一起, 从而获取了迭代核的通项公式 $k^n(s,t)$, 对此无穷求和获得了该类积分方程的解析解, 也就是探论了利用有限项结果通过数学归纳法得到通项公式并最终得到解析解的自动化推理问题; 第四章, 研究了高阶非线性 Volterra-Fredholm 积分 — 微分方程的 Taylor 多项式解的算法, 利用该算法获得了此类积分 — 微分方程的 Taylor 多项式解或解析解. 在此基础上, 研究了所建立的机械化算法用于求解高阶常微分方程的问题. 第五章, 给出了积分方程求解及其机械化算法研究的总结与相关讨论. 为了更清楚展示符号计算算法建立的思想与过程, 将符号积分算法研究及其在微分中值定理自动推证中的应用列于附录中.

本书的编写得到浙江省新世纪 151 人才工程专项基金和温州大学重点学科建设经费的资助.

由于作者水平和能力有限, 书中难免出现不当之处甚至错误, 恳请读者批评指正.

王玮明

二零零八年九月

目 录

第1章 绪 论

积分方程 (integral equation), 即被积函数含有未知函数的方程, 是继微分方程之后出现的一个新的近代数学的重要分支, 它与微分方程、泛函分析、计算数学、位势理论和随机分析有着紧密和重要的联系 [15, 60, 61, 80, 114, 203, 212]. D.Hilbert 曾指出 [48], 积分方程的研究对于定积分理论、级数理论、线性微分方程理论和变分法都是重要的 [60, 80].

众所周知, 数学模型是解决实际问题的一种重要手段. 同一个问题, 既可以用微分方程的定解问题来描述, 也可以用积分方程来描述. 而微分方程的定解问题又可以化为等价的积分方程, 这不仅可以降低维数, 减少计算时所用节点的个数, 缩短计算时间, 还可使未知函数性质上的限制减弱 (只要满足积分方程即可), 这样可以不再通过寻求微分方程所有可能的解到最后再利用定解条件来确定满足定解问题的解, 而是把满足方程与适合定解条件同时体现在积分方程这一紧凑的形式中. 特别地, 积分算子常常较微分算子具有更好的性质, 更适合于分析和论证 [60, 80, 114, 203]. 此外, 有些刻画扩散与迁移现象的数学问题, 不能用微分方程来表示, 为了解决这些问题, 就必须用积分方程或积分 — 微分方程来解决, 如中子迁移理论中的一些问题等 [114].

另外, 积分方程也是科学研究和解决工程技术问题的一个重要数学工具. 在静电学、电动力学、弹性力学、流体力学、电磁场理论、辐射学、地球物理勘探以及航空航天、土木、机械等领域的前沿问题研究中, 许多问题都可化为求解对应的积分方程, 因而有着广泛的应用 [15, 60, 61, 67, 80, 114, 203, 212].

由于 V. Volterra, I. Fredholm 以及 E. Schmidt 等一大批研究人员关于积分方程的出色工作, 使得积分方程这门学科的研究在相当长一段时间内成了一种世界性的狂热, 也使积分方程的研究无论从深度和广度来讲都有了很大的发展, 产生了大量的文献, 而且很多文献具有一定的应用价值. 后来, 由于积分方程的复杂性, 研究的热度有所下降 [60]. 目前, 随着计算机科学与技术的迅猛发展, 为科学界和工程界提供了有力的计算工具, 积分方程及其应用的研究也随之活跃起来, 有关的杂志、会议及出版物也不断涌现, 预示着积分方程及其应用的研究又可能出现新的高潮 [80, 114, 203, 212].

而我们正处在信息时代, 在这个以计算机为标志的革命性时代中, 特别是在吴文俊所创立的数学机械化以及数学机械化在数学的诸多领域成功应用的背景下, 对

于目前相对较为“冷清”的积分方程研究应该有什么样的创新与之相适应呢? 正是基于上述考虑, 本书将结合数学机械化的思想和方法, 展开对积分方程机械化求解的研究.

本章将从积分方程研究简史、积分方程求解方法、数学机械化及其应用等方面进行简要综述.

1.1 积分方程研究概况

根据已有资料, 特别是参考了文献 [15, 60, 61, 80, 114, 203, 212] 中关于积分方程发展简史的有关论述, 本节将综述积分方程研究概况, 并且较详细地叙述在后续几章中涉及到的重要概念或定义, 在此基础上, 本节最后简单归纳总结目前积分方程研究的主要方面.

积分方程如同微分方程一样起源于数学物理问题. 积分方程的一般理论是在 20 世纪逐步发展和成熟起来的. 但是, 积分方程的研究则在 18 世纪末就已经出现了. 早在 1782 年, P. Laplace 就考虑过由

$$f(x)=\int_{-\infty}^{+\infty}e^{xt}g(t)dt \tag{1.1}$$

给出的 $g(t)$ 的积分方程, 而这是今天广泛应用的关于 $g(t)$ 的 Laplace 变换公式. 1823 年, S. Poisson 给出了问题 (1.1) 中 $g(t)$ 的表达式, 即:

$$g(t)=\frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty}e^{xt}f(x)dx \tag{1.2}$$

其中, α 充分大.

19 世纪, 由于科学技术的发展, 从一些实际问题中提出了许多关于积分方程的问题. 例如, 热理论中关于

$$f(x)=\int_{0}^{\infty}\cos(xt)u(t)dt \tag{1.3}$$

的反演问题是真正属于积分方程史的值得注意的结果, 1811 年, J.Fourier 解决了问题 (1.3), 并指出函数

$$u(t)=\frac{2}{\pi}\int_{0}^{\infty}\cos(xt)f(x)dx$$

即为问题 (1.3) 的解.

第一个直接应用并求解积分方程的数学家是 N. H. Abel. Abel 提出了在地球引力场中的一个质点下落问题: 在一个垂直平面内, 求一个质量为 m 的质点在垂直

等加速条件下下落所经的路径, 使其下落的时间等于下落垂直距离 h 的已知函数 $T(h)$.

Abel 从下落质点的动能和势能之间的关系着手研究这一问题. 假设质点下落过程中没有摩擦, 从高度 h 下落到 $y\,(<h)$ 时, 有:

$$\frac{m}{2}\left(\frac{ds}{dt}\right)^2 = mg(h-y),$$

其中, $\frac{ds}{dt}$ 是在 t 时刻的切线速度, s 是弧长, g 是重力加速度. 解出 dt, 并且求积, 即得:

$$\int_h^0 \frac{ds}{\sqrt{2g(h-y)}} = T(h).$$

令 $ds = -u(y)dy$, 即可得下述积分方程:

$$\int_0^h \frac{u(y)dy}{\sqrt{2g(h-y)}} = T(h).$$

在此基础上, Abel 在后续研究中导出了更一般形式的积分方程, 后人称之为 Abel 积分方程:

$$u(x) = \int_a^x \frac{f(y)}{(x-y)^a}dt, \quad (0 < a < 1,\ u(a) = 0), \tag{1.4}$$

其中, $u(x)$ 是已知函数, $f(y)$ 是未知函数. 并且, 他给出了方程 (1.4) 的解为:

$$f(y) = \frac{\sin a\pi}{\pi}\frac{d}{dy}\int_a^y \frac{u'(x)}{(y-x)^{1-a}}dx$$

此后, Abel 方程成为许多数学家研究的对象, 并从多方面加以推广.

J. Liouville 独立于 Abel, 自 1832 年起解决了一些特殊形式的积分方程. Liouville 跨出的最有意义的一步是表明某些微分方程的解可以通过解相应的积分方程而得. 例如, 所要求解的微分方程是

$$y'' + [\rho^2 - \sigma(x)]y = 0, \tag{1.5}$$

其中, $a \leqslant x \leqslant b$, ρ 是参数. 设 $u(x)$ 是满足初始条件

$$u(a) = 1, \quad u'(a) = 0 \tag{1.6}$$

的一个特解, 则 $u(x)$ 必定是非齐次方程

$$y'' + \rho^2 y = \sigma(x)u(x)$$

的解. 应用常微分方程的基本结果, 便有:

$$u(x) = \cos\rho(x-a) + \frac{1}{\rho}\int_a^x \rho(t)\sin\rho(x-t)u(t)dt. \tag{1.7}$$

这样, 如果能够求解积分方程 (1.7), 即可得到二阶微分方程 (1.5) 满足初始条件 (1.6) 的解.

Liouville 得到形如 (1.7) 的解, 用的是被认为属于 C.G.Neumann 的逐次逼近法. 而 Neumann 的涉及讨论逐次逼近法的著作《对数和牛顿位势的研究》(Untersuchungen über das logarithmische und Newton'sche Potential, 1877) 是四十多年后的 1877 年出版的. 逐次逼近法在今天仍是积分方程求解的有效算法之一.

Abel 和 Liouville 所研究的积分方程都属于基本的类型, Abel 研究的是

$$f(x) = \int_a^x k(x,t)u(t)dt, \tag{1.8}$$

而 Liouville 的则是:

$$f(x) = u(x) - \int_a^x k(x,t)u(t)dt, \tag{1.9}$$

在这两种情形中, $f(x)$ 与 $k(x,t)$ 是已知的, 而 $u(x)$ 是待定的函数. 用 Hilbert [48] 引入的今天仍然在使用着的术语来说, 它们分别属于第一种 (the first kind, 未知函数只在方程中的积分号里面出现) 和第二种 (the second kind, 未知函数既出现于方程的积分号下, 也出现在方程的其他地方) 积分方程, $k(x,t)$ 称为积分方程的核 (kernel).

一般地, (1.8), (1.9) 也称为 Volterra 积分方程, 而当上限 x 是固定数 b 时, 它们就称为 Fredholm 积分方程, 即:

$$f(x) = \int_a^b k(x,t)u(t)dt, \tag{1.10}$$

$$f(x) = u(x) - \int_a^b k(x,t)u(t)dt, \tag{1.11}$$

分别称为第一种和第二种 Fredholm 积分方程.

实际上, Volterra 积分方程是相应的 Fredholm 积分方程的特殊情形, 因为在 (1.10), (1.11) 中当 $t > x$ 时取 $k(x,t) = 0$ 即可得相应的 (1.8), (1.9). 而 $f(x) = 0$ 时的第二种积分方程这一特殊情形称为对应的齐次方程.

19 世纪中叶, 积分方程的主要兴趣是围绕着求解 Laplace 方程

$$\Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{1.12}$$

(也称为 “Dirichlet” 问题或 “位势方程”, △ 称为 Laplace 算子) 有关的边值问题而展开的, 方程要在某条曲线 C 所围成的已知平面区域内成立. u 的边值是某一函数 $f(s)$, 它作为沿曲线 C 的弧长 s 的函数给出. 这时位势方程的一个解可以表示成:

$$u(x,y)=\frac{1}{2\pi}\int_C \rho(s)\log\frac{1}{r(s;x,y)}ds,$$

其中, $r(s;x,y)$ 是点 s 到区域内部或边界上任一点 (x,y) 的距离, 而 $\rho(s)$ 是一个未知函数, 它对 C 上的点 $s=(x,y)$ 满足

$$f(s)=\frac{1}{2\pi}\int_C \rho(t)\log\frac{1}{r(t;x,y)}dt. \tag{1.13}$$

显然, 这是关于 $\rho(t)$ 的第一种积分方程. 换一种选择, 如果把方程 (1.12) 的满足同样边界条件的解取作

$$v(x,y)=\frac{1}{2\pi}\int_C u(s)\frac{\partial}{\partial n}\Big(\log\frac{1}{r(s;x,y)}\Big)ds,$$

其中, $\frac{\partial}{\partial n}$ 表示边界上的法向微商, 那么 $u(s)$ 必须满足积分方程

$$f(s)=\frac{1}{2}\phi(s)+\frac{1}{2\pi}\int_C u(t)\frac{\partial}{\partial n}\Big(\log\frac{1}{r(t;x,y)}\Big)dt. \tag{1.14}$$

这是第二种积分方程.

由此可见, 同一个问题既可以用第一种积分方程表示, 也可用第二种积分方程表示. 事实上, 第一种积分方程在一定的条件下可以转化为第二种积分方程.

偏微分方程的另一个问题 — 波动方程, 也是通过积分方程解决的. 1894 年, 大数学家 H.Poincaré 考虑了带复 λ 值的波动方程:

$$\triangle u+\lambda u=f(x,y),$$

在此基础上, 1896 年, 他研究了由上述偏微分方程导出的下列形式的积分方程:

$$u(x)+\lambda\int_a^b k(x,y)u(y)dy=\psi(x),$$

并且断言, 其解是 λ 的亚纯函数.

但值得一提的是, “积分方程” 一词是 1888 年 D. Reymond 第一个提出的.

1884 年起, 罗马的数学物理教授 V. Volterra 在积分方程方面的工作, 是线性积分方程研究的重要转折点, 他也因此被认为是积分方程一般理论的创始人. Volterra 研究了形如 (1.9) 的积分方程, 即第二种 Volterra 积分方程:

$$f(x)=u(x)-\lambda\int_a^x k(x,t)u(t)dt, \tag{1.15}$$

并且证明了: 如果 $k(x,t)$ 在正方形 $a \leqslant x, t \leqslant b$ 上连续, $f(x)$ 在区间 $[a,b]$ 上连续, 则对任意的 λ, 方程 (1.15) 有且只有唯一连续解, 并且该解可用逐次逼近法求得. 此外, Volterra 还给出了第一种 Volterra 积分方程

$$f(x) = \lambda \int_a^x k(x,t)u(t)dt \tag{1.16}$$

的求解方法, 用的方法是将第一种积分方程 (1.16) 化为第二种积分方程 (1.15) 的形式求解. 这种求解方法一直延续至今仍在使用. 在研究过程中, Volterra 还发现, 第一种 Volterra 积分方程能够转化为含 n 个未知数的 n 个线性代数方程组的形式 $(n \to \infty)$.

与此同时, 斯德哥尔摩的数学教授 I. Fredholm 也一直潜心于 Dirichlet 问题的求解, 他吸收了这一时期关于积分方程的各种论述和研究思想, 系统地研究了积分方程理论. 1899 年, 他在给老师 Mittag-Leffler 的一封信中提出了如下形式的积分方程:

$$f(x) = u(x) + \lambda \int_0^1 k(x,t)u(t)dt, \tag{1.17}$$

其中, $k(x,t)$ 是正方形 $0 \leqslant x, t \leqslant 1$ 上的已知连续函数, $f(x)$ 在区间 $[0,1]$ 上连续, 而 $u(x)$ 是未知函数. Fredholm 认为, 积分方程 (1.17) 的解可以表示为 λ 的两个整函数的商的形式. 1900 年, Fredholm 发表了第一篇关于积分方程求解理论的论文 [32], 提出了与核函数 $k(x,t)$ 有关的 Fredholm 行列式 $d(\lambda)$ 和 Fredholm 子式 $D_\lambda(x,t)$, 并且证明了它们都是 λ 的整函数. 在该文中, Fredholm 还巧妙地证明了: 如果 λ 是函数 $d(\lambda)$ 的一个零点, 则积分方程 (1.17) 的齐次方程

$$u(x) + \int_0^1 k(x,t)u(t)dt = 0$$

有不恒等于 0 的解. 此即为著名的 Fredholm 第二定理.

1903 年, Fredholm 又发表了另一篇论文 [33], 对积分方程 (1.17) 作了进一步的研究, 他取 $\lambda = 1$, 得到了如下结果: 如果 $d(1) \neq 0$, 则有且仅有一个函数 $u(x)$ 满足积分方程 (1.17), 并可表示为:

$$u(x) = f(x) - \int_0^1 \frac{D_1(x,t)}{d(1)} f(t)dt.$$

此即 Fredholm 第一定理.

Fredholm 得到的关于积分方程的很多结果, 极其密切地平行于齐次与非齐次线性代数方程组的理论.

由于 Volterra, Fredholm 等人关于积分方程的出色工作, 使得积分方程这一研究领域吸引了许多数学家的注意, 尤其是 D. Hilbert 在积分方程理论的宏伟工作.

从 1904—1910 年, Hilbert 在《拉丁根自然科学皇家学会报告》(Nachrichten der Königlichen Gesellschaft der Wissenschaften zu Göttinggen) 上一连发表了 6 篇论文, 后收录于其专著《线性积分方程一般理论的原理》(Grundzüge einer allgemeinen Theorie der linearen Integralgleichungen, 1912) [48] 中, 系统研究了线性积分方程的一般理论, 并且把积分方程应用到各种几何和数学物理问题的解决中.

Hilbert 关于积分方程的工作有许多重大成果, 其中包括, 他用定义完备正交系这一重要概念把无限多个变量二次型理论运用到积分方程; 关于对称核建立了一般的谱理论; 建立了任一连续函数展开成核的特征函数的级数理论 (Hilbert-Schmidt 定理); 引进了双线性形式, 开创了双线性对称形式的谱理论; 证明了一个函数展成第二种积分方程的特征函数的展开式取决于相应的第一种积分方程是否可解; 等等. 而他最有价值的成就之一是把微分方程的 Sturm–Liouville 边值问题化成积分方程的创举. Hilbert 证明了

$$\frac{d}{dx}\left(p(x)\frac{du}{dx}\right)+q(x)u+\lambda u=0 \tag{1.18}$$

满足边界条件

$$u(a)=0,\quad u(b)=0$$

(甚至更一般的边界条件) 的特征值和特征函数, 恰恰是积分方程

$$u(x)-\lambda\int_a^b G(x,t)u(t)dt=0 \tag{1.19}$$

的特征值和特征函数, 其中 $G(x,t)$ 是方程 (1.18) 的 Green 函数, 即

$$\frac{d}{dx}\left(p(x)\frac{du}{dx}\right)+q(x)u=0$$

的特解, 它满足一定的可微条件, 并且它的一阶偏微商在 $x=t$ 有等于 $-\frac{1}{p(t)}$ 的奇性跳跃. 类似的结果对偏微分方程也成立. 据此, 积分方程变成了求解微分方程的一种方法, 这种研究方法被成功地用于解决愈来愈多的物理问题. 后来, 人们还发现, n 阶常微分方程的边值问题、偏微分方程问题的研究都可归于某个积分方程的研究. 从而, 掀起了积分方程的理论及其应用研究的世界性狂热.

Hilbert 在积分方程方面的工作, 被在德国几所大学担任数学教授的 E. Schmidt 简化了, 他用的方法是 H. Schwarz 在位势理论研究中创立的. 他最有意义的工作是在 1907 年把特征函数的概念推广到带非对称核的积分方程.

匈牙利数学教授 F. Riesz 在 1907 年继续了 Hilbert 的工作, 特别是创造性地引进了 Lebesgue 平方可积函数的定义, 即二重 Lebesque 积分

$$\int_a^b |k(x,y)|^2dxdy<\infty$$

存在. 在此基础上, Riesz 证明了如果 $f(x)$ 与 $k(x,t)$ 是 Lebesgue 平方可积的, 则第二种 Fredholm 积分方程 (1.11) 是可解的, 并且除了一个在 $[a,b]$ 上 Lebesgue 积分为 0 的函数外, 解是唯一的. 而 Riesz 1910 年发表在《Mathematische Annalen》中的论文被认为是泛函分析核心的抽象算子研究的开端. 在该文中, 他把第二种 Fredholm 积分方程 (1.11) 的解推广到 L^p 空间中的函数, 并把 Fredholm 积分算子

$$\lambda \int_a^b k(x,t)u(t)dt$$

设想成作用在某一 Banach 空间中的全连续算子, 而未知函数 $u(x)$ 与已知函数 $f(x)$ 是该空间中的点. Riesz 的这一结果, 后来被 J. Schauder 作了补充, 并发展成为泛函分析中关于全连续算子的 Riesz-Schauder 理论.

与此同时, 科伦大学数学教授 B. Fischer 引进了平均收敛的概念, 即当

$$\lim_{n\to\infty} \int_a^b \Big(f(x) - f_n(x)\Big)^2 = 0$$

时, 称函数序列 $\{f_n\}$ 在区间 $[a,b]$ 上平均收敛到唯一确定的函数 $f(x)$. 这里的积分是 Lebesgue 积分. 在区间 $[a,b]$ 上, Lebesgue 平方可积的函数集合记为 $L^2[a,b]$, 或简记为 L^2. Fischer 的主要结果是说, $L^2[a,b]$ 在平均收敛意义下是完备的, 即: 若 $f_n \in L^2$, 并且 $\{f_n\}$ 平均收敛, 那么在 L^2 中存在一个函数 $f(x)$, 使得 $\{f_n\}$ 平均收敛到 $f(x)$. 这种完备性是平方可积函数的主要优点. 另外, Fischer 还强调 Lebesgue 平方可积函数的运用是本质的, 没有更小的函数集合可用.

差不多紧接着 Fredholm 积分方程理论的出现, 与它本质上完全不同的奇异积分方程理论也随之产生了, 这类积分方程是人们从不同角度研究不同问题而提出的. Hilbert 在研究解析函数的某些边值问题时发现了它们 [48], 几乎同时, Poincaré 在研究潮汐现象时也发现了它们 [104]. 这类具有代表性的奇异积分方程是下列形式的积分方程 [15]:

$$A(t)\varphi(t) + \frac{1}{\pi i}\int_L \frac{k(x,\tau)}{\tau - t}\varphi(t) = \eta(t),\ t \in L, \tag{1.20}$$

其中, L 是平面上的光滑曲线, 系数 $A(t)$, 自由项 $\eta(t)$ 和核 $k(x,t)$ 都是曲线弧 L 上按 Holder 意义连续的已知函数, $\varphi(t)$ 是未知函数. 积分方程 (1.20) 在一定条件下是按 Cauchy 主值的意义存在, 故也称之为 Cauchy 核积分方程. 而在研究大气辐射传输问题时, 提出了另一类积分区域是无穷的奇异积分方程, 其代表性的例子是 Wiener-Hopf 方程:

$$\varphi(x) - \int_0^\infty k(x-y)\varphi(y)dy = \psi(x),\ 0 \leqslant x < \infty, \tag{1.21}$$

其中, $k(x-y)$ 与 $\psi(x)$ 是已知函数. 这类奇异积分方程的理论也有很大发展, 并且在许多实际问题中有着重要的应用 [15].

另一方面, 由于实际发展的需要, 非线性积分方程的研究也出现了不少结果, 然而, 对一般的非线性积分方程还缺乏系统的理论, 甚至于方程可解性的讨论也比较困难 [44].

随着泛函分析理论的发展和完善, 人们将微分方程和积分方程的研究置于抽象空间的框架之下, 使积分方程的理论更加完善, 应用也更加广泛, 同时也为微分 — 积分方程的研究奠定了基础.

在中国, 最早从事积分方程研究的老一辈专家和学者有张世勋、陈传璋、张石生等. 随着中国各项科学事业的发展, 积分方程也得到相应的发展, 参与工作的新生力量日益增多, 研究的范围几乎涉及积分方程的各个领域. 目前, 许多高校开设了积分方程理论及其应用研究的课程, 这将为积分方程的研究注入新的活力.

虽然, 由于积分方程本身的复杂性, 关于积分方程的研究不再像 20 世纪初期那么狂热, 特别是, 直到现在, 关于第一种 Fredholm 积分方程 (1.10) 尚未建立起系统的理论, 正如从积分方程 (1.10) 本身的构造可看到, 即使方程 (1.10) 的核 $k(x,y)$ 是退化核这种最简单的情形, 此方程也并不总是有解的. 但就积分方程目前主要研究, 可简单归纳为沿着如下七个方面进行的 [212]:

第一个方向在于揭示新的积分方程类, 以及其成立的线性代数方程组的基本定理, Fredholm 关于特征值的分布定理等;

第二个方向是与正交分解和对称核理论相关联的, 特别是 Hilbert 空间中的算子理论;

第三, 研究经典 Fredholm 定理不成立的线性积分方程, 即所谓的奇异积分方程, 如 Cauchy 奇异积分方程、Wiener-Hopf 方程等;

第四, 研究非线性积分方程可解性及求解;

第五, 研究随机积分方程可解性及求解;

第六, 研究各类积分方程的数值解法;

第七, 研究各类积分 — 微分方程可解性及求解.

另外, 关于积分方程在各个领域的应用研究也始终贯穿于积分方程的发展中.

1.2 积分方程求解方法概述

对自然科学中很多问题的研究大致可以分为两大类: 一是定性研究, 二是定量研究. 对于积分方程研究而言, 定性研究即为可解性条件判定, 定量研究即为方程的求解.

积分方程求解是一个古老且在理论和实际中都很重要的研究课题. 积分方程解, 即解析解 (analytic solution, 也称为精确解) 和数值解 (numerical solution), 可以很好地解释各种物理现象, 有助于人们展开对积分方程所刻划的现实物理现象的分析和研究.

上节谈到, 第二种 Fredholm 积分方程 (1.11) 已经建立了系统的理论 (即 Fredholm 理论), 对于第一种 Fredholm 积分方程 (1.10) 到现在尚未建立起系统的理论; 而第一种 Volterra 积分方程 (1.19), 在一般情况下, 可以通过求微商的方法化为第二种 Volterra 积分方程 (1.18). 基于此, 本节将简单归纳积分方程的求解方法及其研究概况, 不详细介绍各种解法的具体步骤, 而对于本书研究中用到的算法将在后续章节中详述.

对于积分方程解的研究大概可以分为以下三个方面:

其一, 在难以求得解析解的情况下, 对解的适定性 (存在性、唯一性、稳定性等) 进行分析;

其二, 借助于计算数学理论和计算机技术, 获取积分方程数值解或近似解;

其三, 应用某些数学技巧或假设, 构造适当的变换使方程简化并求出解析或近似解 (approximate solution).

如同微分方程一样, 由于积分方程本身的复杂性, 已有的大量的积分方程无法获得解析解. 于是, 人们转而设法去求解积分方程的近似解, 并为此发展了许多可行的求解方法.

积分方程求解方法虽然名目繁多, 但通常可以将这些方法分为两类:

一类是对解析解作近似逼近, 例如:

(1) 逐次逼近法 (method of successive approximations) [15, 61, 80] 等;

(2) 迭代法 (iteration method) [2, 13, 15, 61, 80, 81, 100, 203, 217] 等.

这类方法属于经典的应用广泛的求解积分方程的方法, 目前, 单一地使用这些方法显现出很多局限性, 大多数情况下, 人们选择与其他方法混合使用, 已成功地解决了许多方程求解问题.

另一类是化为便于进行数值计算的其他类型的问题, 主要方法有: 将核用退化核代替 (此方法需要经过复杂的数学分析过程), 或者用数值积分公式将积分方程化为代数方程组, 或者把积分方程化为变分问题, 或者通过构造正交基运用待定系数法求解, 等等. 这类方法中很多是近年来发展起来的新算法, 在积分方程求解中具有重要作用. 常见的方法主要有:

(1) 积分法 (quadrature method) [7, 8, 27, 51, 61, 85, 86, 89, 114, 204, 212, 216];

(2) 投影法 (projection method) [2, 61, 77, 106];

(3) 插值法 (interpolation method) [2, 47, 56, 61, 95, 108, 154, 183, 140, 155];

(4) 级数法 (series method), 包括 Taylor 级数法 [61, 93, 94, 110, 184, 185], Walsh 级数法 [111, 119] 等;

(5) 配置法 (collocation method) [6, 11, 14, 45, 52, 57, 61, 77, 99, 198, 197], 以及在此基础上发展的混合插值配置法 (mixed interpolation collocation methods) [9, 10], 级数展开配置法 (Sinc-collocation method) [98, 109], 样条配置法 (spline collocation method) [1, 50, 58, 95, 101, 140];

(6) Tau 方法 (tau method) [49, 102, 105, 107, 112, 214];

(7) Galerkin 法 (Galerkin method) [43, 46, 74, 94, 103, 179, 211];

(8) Adomain 分解法 (Adomain's decomposition method) [3, 5, 151, 152];

(9) 并行算法 (parallel method) [12, 13, 91, 121],

(10) 外推法 (extrapolation method) [51, 75, 88, 126];

(11) 荣格 — 库塔法 (Runge-Kutta method) [4, 90, 201],

(12) 小波法 (wavelet mathod) [43, 74, 78, 82, 92, 113, 123, 124, 179, 200];

(13) 神经网络算法 (neural network) [28, 42, 53, 125, 186, 196, 215], 等等.

详细叙述这些求解方法将会使本书过于冗长且并非本书重点, 故在此不再赘述. 上述求解方法均可在相应的参考文献中查到详细的算法描述及应用举例.

通过上述求解方法, 人们获得了各种积分方程诸多形式的精确解或者近似解. 但是, 由于积分方程求解问题本身的复杂性, 利用这些方法求解时不可避免地遇到冗长繁复、令人望而生畏的各种计算, 且传统的纸笔演算耗时、费力又易出错, 更重要的是积分方程求解过程中出现的超大计算量的人力难以胜任的十分复杂且精确的代数与微分、积分等符号计算以及"中间过程膨胀"等问题, 使得积分方程求解成为该领域研究的难点之一.

所以, 借助计算机的大容量、高速度的特点, 用精确的符号计算, 机械化地实现积分方程求解具有十分重要的意义和广泛的应用前景.

下节我们将讨论数学机械化的若干问题, 为机械化求解积分方程奠定基础.

1.3 数学机械化简介

1.3.1 数学技术

数学是一切科学与技术的基础, 是一切关键技术中的关键. 当代数学的一个重要特征和发展趋势, 是数学内部各分支学科的高度发展和相互之间在内容、概念及方法上的不断交叉和融合, 充分地显示出数学是一个密不可分的整体. 数学与其他学科以及整个外部世界的联系、交叉、渗透与融合也不断得到加强 [63, 118, 195]. 现在看来很冷清的学科, 将来某一天可能会发现重要的应用, 变成很热门的领域. 例

如, 像近世代数这样很抽象的学科, 现已成为密码学研究最重要的一个工具.

更重要的是, 数学的应用已经拓展到几乎每个科学领域和应用部门, 而且在其中起着关键的不可替代的重要作用 [63]. 目前, 数学应用的范围逐渐扩大, 已从以往传统领域 (如力学、物理、天文以及传统工业领域) 扩展到非传统的领域 (如化学、生物、其它各门自然科学及高新技术领域), 甚至进入经济、金融、保险及很多社会学的领域, 深入到各行各业, 可以说是无所不在, 且发挥着越来越重要的作用.

近三十多年来, 计算机科学技术的飞速发展对数学科学产生了巨大的影响, 各种高性能的算法日益得到重视和发展. 现在, 人们已由利用计算机自编程序算题, 到应用现成的大规模的通用软件, 到当今的虚拟现实、临境技术, 进展的速度令人目不暇接. 计算机的迅速发展和普及, 不仅为数学应用提供了强大的技术手段, 而且极大地改变着数学的研究方法和思维模式. 数学与计算技术的结合已形成一种关键性的、可实现的技术, 称为 "数学技术" [63, 195]. 在这种技术中起关键作用的部分是数学, 拿走它就只剩下一个空壳, 而这种技术由于计算机科学技术的进步可以是即时的、在线的, 是真正可以实现的. 例如, 一个三维物体, 如果知道它在各个方向的平面投影, 就能推算出它的精确形状. 这是一个数学上的问题, 已由拉氏变换所解决. 现在的 CT 技术, 本质上就是通过按各个不同方向的 X 光投影来重现人体内部肿瘤的位置及形状, 完全是同一个问题, 所以 CT 仪器的核心是体现拉氏变换的计算机软件及硬件, 舍此则一文不值. 因此, 数学技术本质上是数学的内容物化为计算机软件及硬件, 成为技术的一个重要组成部分. 这样, 数学技术变成了产品、而且是高科技产品, 真正、直接转化为先进的生产力, 从而产生巨大的经济效益, 同时, 也进一步体现了数学的价值 [63].

由于现代技术愈来愈向高级精密发展, 定量方面要求非常精细严密, 现在的高技术本质上是一种数学技术, 这一观点已为越来越多的人所认同, 而且已出现不少有效的数学技术. 显而易见, 这将成为今后一个重要的发展方向.

1.3.2　计算机数学与计算机代数系统

计算机科学技术的迅猛发展对数学科学产生了巨大的冲击, 对数学研究的观念及研究方法都产生了深刻的影响. 引发这场冲击波的最主要的数学事件是 "四色猜想"(Four-color conjecture) 的证明. 1976 年 9 月, 美国《Bulletin of the American Mathematical Society》宣布伊利诺大学的两位数学家 Appel 和 Haken 解决了困扰数学界长达近两百年的 "四色猜想", 这一成果震惊了整个数学界, 因为两位数学家的论证中有很大一部分并且是关键的部分是由计算机完成的, 而整个证明过程涉及的超大量的运算, 对其进行逐步人工检验是不可能的. 这就意味着 "数学证明" 的概念发生了突变 [54]. 因为在传统上, 数学问题只能通过严密可靠的数学推理才能得到证明, 但 "四色猜想" 的结果却是借助计算机程序而得, 因而在当时有一大批数学

家不承认这是一个数学证明. 经过长时间的争论, 这一结果最终还是被数学界接受了, 并对数学研究方法产生了革命性的影响 [141].

另一方面, 科学计算始终是数学研究的重点和难点, 也是数学应用的前提. 科学计算包括两类: 一类是数值计算, 例如求函数值、方程的数值解等; 另一类是符号计算, 又称代数运算, 这是一种智能化的计算, 处理的是符号, 符号可以代表整数、有理数、实数和复数, 也可以代表多项式、函数, 还可以代表数学结构如集合、群等 [137, 138, 141].

科学计算常常是一项非常繁琐的工作, 事实上, 人类很早就意识到在解决实际问题时对于数学特别是 "计算" (包括数值计算、逻辑运算、符号计算、图形计算等) 的极端重要性, 并致力于 "计算" 的机械化及计算机械的发明创造 (如算盘, 对数计算器、计算尺、加法机、Leibnitz 演算机、差分机, 等等). 直到计算机的出现和发展才逐步解决了数值计算中的困难. 从计算机发明到现在六十多年间, 用计算机进行的科学计算主要是数值计算, 如天气预报、油藏模拟、航天等.

而用计算机进行符号计算是数学和计算机领域的一个新的发展方向. 长期以来, 数学家和计算机科学家梦想用计算机代替人脑进行代数符号运算以及数学的各种处理, 使数学走向 "机械化" 的道路, 从而也使计算机本身更加智能化. 中国著名数学家吴文俊院士首先提出的 "吴方法" 为数学定理证明在计算机上的实现奠定了理论基础 [156]–[178], 并在几何定理机器证明、方程组求解、微分几何、理论物理、机器人学、计算机图形学等数学和高科技领域相继获得了广泛的应用 [39].

20 世纪 80 年代以来, 用计算机进行符号计算的研究在国内外发展非常迅速, 涉及的数学领域不断地扩大, 出现了多种符号运算方法、计算程序和系统, 如符号运算、符号处理、计算机代数等等. 这是一个以构造性数学为核心, 以计算机实现为目标, 以实用算法为研究内容, 以实用程序或软件为成果的研究领域, 这个领域被称为计算机数学 (computer mathematics), 也称为计算机代数 [137, 138, 141]. 计算机数学的发展逐步产生了一些独立的计算机程序库, 称为计算机代数系统 (computer algebra system).

计算机数学是数学研究领域的新方向, 是相对年轻的学科, 研究文献散布在各种涉及数学计算的杂志上. 目前这一学科已有专门的研究杂志, 如: Journal of Symbolic Computation (Elsevier Press) 是发表计算机代数方面研究成果的主要国际期刊. 还有 Applied Mathematics and Computation (Elsevier Press), Applicable Algebra in Engineering 和 Communication and Computing (Springer-Verlag). 除以上杂志外, 从大量的会议论文集中可以查阅到各种计算机代数的研究成果和研究趋势. 另外, 美国 Association for Computing Machinery (简称 ACM) 支持的学术研讨会 International Symposium on Symbolic and Algebraic Computation (ISSAC) 每年夏天在欧洲和其他国家或地区举行, 并由 Springer 出版会议论文集. 在亚洲各国举办

的 Asian Symposium of Technology on Mathematics (ASTM) 和 Asian Symposium on Computer Mathematics (ASCM) 也逐渐成为国内外计算机数学工作者学术交流和报告研究成果的一个园地.

而计算机数学的产品 — 计算机代数系统 (也称数学软件) 在不断地发展、提高和完善. 计算机代数系统为数学的教学、研究和应用开辟了新天地, 既使数学实验成为可能, 又使数学成果直接为实际服务以及使数学的大众化成为可能.

根据计算机代数系统的用途, 将其分为两类: 专用系统和通用系统. 专用系统主要是为解决物理和数学某些分支中的特殊问题而设计的, 专用系统在符号和数据结构上都适用于相应的领域而且多数是用低级语言编写的, 因此专业人员使用很方便, 计算速度也较快, 它们在专业问题的研究中起着重要的作用; 通用系统具有多种数据结构和丰富的数学函数, 应用领域广泛.

目前, 一部分计算机代数系统发展成为完整的专用或通用的计算机数学软件, 如美国的Mathematica, Matlab, 加拿大的Maple 等 [138, 141], 以及以高小山为首席科学家领导研发的独立、完整、高效、具有自主知识产权的数学机械化平台 MMP [40], 它们为科学研究、工程应用、教学中的计算和推理提供了一个强有力的应用和开发平台. 这些通用计算机代数系统大都是基于 C 或者 C^{++} 语言研发的.

其中, Maple 系统的研发始于 20 世纪 80 年代. 1980 年 9 月, 加拿大 Waterloo 大学的符号计算机研究小组成立, 开始了符号计算在计算机上实现的研究项目, 数学软件Maple能是这个项目的产品. Maple 的第一个商业版本是 1985 年出版的. 随后几经更新, 到 1992 年, Windows 系统下的Maple 2 面世后, Maple 被广泛地使用, 得到越来越多的用户. 特别是 1994 年, Maple 3 出版后, 兴起了Maple 热. 1996 年初, Maple 4 问世, 1998 年初, Maple 5 正式发行. 目前广泛流行的是 2006 年面市的Maple 11. 2008 年, 又推出了Maple 12 [96]. 当然, 这仍是一个正在研究中的项目.

Maple中有 3000 多个用于科学计算的函数, 是一个具有强大符号运算能力、数值计算能力、图形处理能力的交互式计算机代数系统. 它可以借助键盘和显示器代替原来的笔和纸进行各种科学计算、数学推理、猜想的证明以及智能化文字处理. Maple 这个超强科学计算工具已经被广泛用于数学、密码学、控制论、物理学、生物学、商学、经济学和工程技术, 它不仅适合数学家、物理学家、工程师, 还适合化学家、生物学家和社会学家, 也是众多高等院校科学和工程实验室的标准科研与教学工具, 它的用户遍及全球 [137, 141]. 总之, 它适合于所有需要科学计算的人.

本书关于积分方程求解的机械化算法研究正是基于Maple平台实现的.

1.3.3 数学机械化

计算机数学研究领域一个重要内容是数学机械化 (mathematics mechanization).

正如吴文俊院士所言: 不论是机器代替体力劳动, 或是计算机代替某种脑力劳动, 其所以成为可能, 关键在于所需代替的劳动已经“机械化”. 所谓机械化, 简单地说, 就是“刻板化”或“规格化”. 数学问题的机械化, 要求在运算或证明过程中, 每前进一步之后, 都有一个确定的、必须选择的下一步, 这样沿着一条有规律的、刻板的道路, 一直达到结论 [176].

回顾数学发展史, 主要有两种思想: 一是公理化思想, 另一是机械化思想. 前者源于古希腊, 后者源于中国古代数学, 这两种思想对数学发展都曾起到巨大作用 [39]. 据此, 可以简单地将数学研究分为两类: 存在性数学与构造性数学. 存在性数学注重研究数学对象的存在性以及它们的性质和其间相互关系, 它是近代数学发展的主流 [138]. 构造性数学退居其次是有特定历史根源的, 即使很基本的数学对象 (例如 4 次代数方程的根式解), 它们的构造也是非常困难和复杂的. 传统的纸笔演算需要消耗数学工作者大量的时间和精力, 因而使其无法从事更具创造性的工作. 随着数学的纵深发展, 与之相应的推理和计算也愈来愈复杂, 而且其中的大量计算又机械乏味. 在这种情况下, 数学家就不得不有所取舍, 放弃对数学对象的构造性而去研究它们的存在性.

但是, 构造性为数学的应用提供支撑, 也是数学机械化的基础. 这并非要低估存在性数学的重要价值和意义, 但过分追求数学的理论发展与优美而不顾其实用性似乎有失偏颇. 对于数学机械化而言, 只证明数学问题解的存在性是远远不够的, 必须有具体、确定的算法把解构造出来, 而当解的具体形式构造出来后, 其存在性也就不证自明了. 就此而言, 构造性数学比存在性数学要困难得多.

事实上, 数学的存在性与构造性之间的争论由来已久, 中国古代数学的发展向来以构造性为特色, 重方法和实用, 这种具有中国特色的构造性数学正在当代数学家吴文俊院士的倡导下走向复兴 [39, 117, 178].

实现数学的机械化是实现脑力劳动机械化的重要理论基础. 数学为其他学科提供了定量化描述问题的语言与解决问题的有效方法, 是科学技术的重要理论依据. 正是由于数学的这种基础性, 每个时代都有与之相适应的数学. 为利用计算机的强大计算能力, 数学的很大一部分内容正在转变为计算机可以接受的语言. 具体地讲就是数学的离散化、算法化和软件化 [39, 40, 176].

正是在这样的背景下, 吴文俊院士在 20 世纪 70 年代末提出了数学机械化的设想, 可以概括为“数学机械化纲领” [40]:

(1) 在数学的各个学科选择适当的范围实现机械化, 推动数学发展与脑力劳动机械化;

(2) 应用数学机械化方法解决相关高科技领域的关键基础理论问题.

现阶段数学机械化研究主要关注的是科学研究与高技术应用中经常遇到的两类问题: 方程求解和自动推理. 事实上, 方程求解已经成为当今诸多重要科学领域

与高新技术研究的核心问题. 数学机械化方法的核心是方程求解的“吴方法”, 这一方法不仅适用于代数方程, 而且适用于微分方程、差分方程、不等式系统等.

简单地说, 数学机械化的基本内容主要有两方面: 一方面, 将传统数学中可以算法化求解、求证的问题利用计算机及数学软件完成或实现; 另一方面, 实现数学机械化首先要将问题转化为可算法化问题, 然后按照 MAP 思想, 即 Model (模型)+Algorithm (算法)+Program (程序), 确定正确的具有操作性的算法, 再选择适当的平台设计相应的计算机程序实现. 也就是:

问题 → 数学问题 → 算法 → 程序 → 计算机实现

因此, 数学机械化的基础是算法设计和程序设计. 所谓算法设计, 就是依据已有的或者发展新的数学理论, 提出有效的方法, 将这些方法描述为适合实施的算法, 并证明所设计算法的正确性和终止性. 算法设计的基础是构造性数学, 辅之以逻辑与算法理论. 这里, 重点是数学对象的构造而非存在性证明, 并且数学的理论和方法需要更加严密, 而逻辑学与算法理论则有助于将这些数学理论和方法形式化和算法化.

数学机械化理论和方法的建立, 是深层次的知识创新, 是适应信息时代产生的基础研究领域, 机械化数学是数学的一部分. 随着计算机大规模的渗透人们的生活, 自然也改变了人们的学习、工作和从事研究的方式, 一张纸、一支笔做数学的情景基本上已成为历史. 机械化数学的努力目标就是要将数学的各个领域一部分一部分的机械化, 从而使传统数学的许多方面由于有了数学机械化而面貌一新.

在吴文俊院士开创性工作的影响下, 经过国内外专家、学者近三十年的努力, 已经成功地将吴方法用于解决力学 [172]、物理 [163]、化学 [170] 等传统领域及机器人 [166]、几何造型 [176]、连杆设计 [165, 176] 等高科技的问题. “吴方法” 还被用于多项式因式分解 [136, 138], 发现微分系统新的极限环 [128, 132, 84], 求解微分方程的行波解与孤立子解 [29, 69, 70, 73], 理论物理 [115, 116], 几何造型中的曲面形式转换问题 [16, 40], 逻辑公式的证明 [153], 计算机视觉 [59, 181], 连杆设计问题 [38, 40], 智能 CAD [24], 组合恒等式自动证明 [18], 等, 逐渐形成了具有中国特色的数学机械化和自动化推理理论, 不仅成功应用于一些相关科学分支, 且成功地解决了一系列高新技术领域中的关键基础理论问题, 获得了国内外学术界的高度称赞和广泛重视.

在机器证明方向, 吴文俊提出的用计算机证明几何定理的方法 (国际上称为 “吴方法”), 首次实现了几何定理高效的自动证明, 显现了无比的优越性. 吴文俊院士的工作被称为自动推理的先驱性工作, 并于 1997 年获得 “Herbrand 自动推理杰出成就奖”.

众所周知, 微分几何、代数几何是引人入胜的数学分支, 它们不但在理论的发展长河中考验了一大批杰出的数学家, 在许多工程应用中也起到了不可替代的作用.

代数几何中, 在等价的意义下做分类, 是非常重要且基本的问题. 利用 "吴方法", 在一维情形下, 用机械化数学的方法, 做出了同构意义下的分类处理, 是值得继续扩展成果的方向 [55]. 其后, 张景中、周咸青、高小山等人又提出了基于几何不变量的 "消点法", 用吴方法给出了超越函数的证明算法, 并给出了 Euclid, Minkowski, Laboshevski, Riemann 等九种几何的分类 [20, 21, 22, 35, 36, 37, 23, 24, 207], 张景中、杨路、高小山、周咸青提出了几何定理机器可读性证明 [206, 193, 194, 209], 李洪波等人将 Clifford 代数与重要的 E.Cartan 外微分运算相结合, 提出了基于 Cliford 代数与吴方法的向量算法, 形成了局部微分几何定理的机器证明的新算法 [65], 利用这一结果可以给出陈省身关于曲面论中一个十分深刻的定理的非常简单的证明, 王东明等人对这一方法作了进一步的改进, 使得这一算法不仅可以证明几何定理, 还可以证明微分几何定理 [127, 133, 130, 131, 134]; 林东岱和刘卓军将吴方法推广至有限几何 [76]; 杨路、曾振柄、夏壁灿等人研发的用于不等式自动推证的 BOTTEMA 以及在此基础上改进研发的 DISCOVERER [120, 189, 190, 192, 191]; 陈永川等人关于组合数学的机器证明 [18]; 吴尽昭等人的关于逻辑及非逻辑定理的机器证明 [153]; 等等.

在非线性方程组求解方向上, 吴文俊院士提出的 "吴方法" (也称 Ritt-Wu 方法) 与结式消元法、Gröbner 基消元法同为国际通用的三大最完整的消元算法 [136], 这是数学机械化的核心. 中国很多学者在这一领域取得了出色的成果. 杨路、候晓荣、曾振柄提出了对任意次数的实系数多项式建立完全判别系统的一个通用算法 [34, 188], 使这个实代数的问题得到圆满的回答; 张景中、梁松新解决了复系数多项式完全判别系统的问题 [208]; 陆征一等人提出了多项式实根分离算法并将其应用于微分方程小扰动极限环的构造 [84]; 王东明基于吴方法研发了用于多项式消元和分解的应用软件 Epsilon [135]; 等等.

非线性微分方程的解, 不仅仅是微分方程理论中关心的重要问题, 同时它还具有十分明显的物理应用背景, 一些著名方程, 如力学, 固态物理、等离子体物理和化学物理等领域中出现的一类非线性波动方程, 需要求物理上有兴趣的钟状、纽结状的孤立波解. 国内外许多专家、学者在此领域作出了卓越成就, 如 Parkes 小组基于 Mathematica 的双曲正切法软件包 ATFM, Hereman 小组的较完整的求非线性演化方程的软件包 PDESS [182], 等等. 国内代表性的成果有李志斌小组关于非线性行波解、孤立子解的系列成果 [69, 70, 71, 72, 73, 79, 210, 182, 199], 张鸿庆小组提出的 CD 对解分法以及系列成果 [17, 19, 62, 64, 87, 97, 122, 180, 187, 202, 205, 213], 石赫 [115, 116]、范恩贵 [29, 30, 31] 等小组的研究成果. 目前, 已经应用吴特征列方法解决了几十个非线性波动方程和非线性发展方程, 所得结果除涵盖已知解外, 还发现了许多新解. 李子明、M.Singer、吴敏与郑大彬关于计算 Laurent-Ore 模的一阶子模的递归方法的研究 [68], 开启了关于多变量函数机械化处理的算法研究, 并获国际

计算机科学协会 (ACM) 符号与代数计算专业委员会颁发的 2006 年度“ISSAC 杰出论文奖”. 这是中国学者首次获得该项奖励.

总之, 吴文俊院士所倡导的数学机械化研究, 一方面继承了古代中国数学思想的精华, 另一方面适应了现代科学技术的发展. 随着时间的推移、工作的积累和方向的拓展, 数学机械化必将为中国乃至世界数学的发展做出积极的贡献, 也必将使数学更好地为科学技术服务, 尤其是为高科技提供理论武器和有效的工具.

前文已经述及, 积分方程求解是积分方程研究的重点和难点. 但是, 积分方程机械化求解的文献尚不多见. 基于上述种种认识和思考, 自 2004 年始, 本书作者以国际通用计算机代数系统Maple 为平台 [141], 在诸多专家和学者研究工作的基础上, 利用数学机械化的思想和方法研究了积分方程求解以及机械化算法设计等问题, 取得了一系列成果 [144, 145, 146, 147, 148, 149, 150]. 本书的核心内容正是这些研究工作的整理与总结.

虽然, 这些研究显得有点稚嫩, 但窃想这将为积分方程求解研究注入新的活力.

第2章　Fredholm 积分方程求解及其机械化

Fredholm 积分方程是一类经典且应用广泛的积分方程, 关于该类积分方程的求解研究已经持续了一百多年 [60]. 尽管如此, 到目前为止, 关于第二种 Fredholm 积分方程 (1.11) 已经建立了系统的理论 (即 Fredholm 理论), 而对于第一种 Fredholm 积分方程 (1.10) 到现在尚未建立起系统的理论. 因此, 本章重点关注第二种 Fredholm 积分方程的机械化求解问题.

本章主要内容包括 3 节, 第 1 节是 Fredholm 积分方程的求解及其机械化算法研究, 第 2 节, 我们探讨了 Fredholm 积分方程组的求解及其机械化算法, 并分别通过大量的算例展示了机械化求解 Fredholm 积分方程 (组) 的正确性以及高效性. 最后, 第 3 节, 给出了本章的结论及进一步的讨论.

2.1　Fredholm 积分方程求解

第一章 1.2 节已经简单归纳了 Fredholm 积分方程求解方法, 在此, 不再赘述. 除了前已述及的那些方法外, 求解第二种 Fredholm 积分方程还有一种方法, 即豫解核 (resolvent) 求解法 [61, 212]. 这是一种经典的、由 Fredholm 建立的求解方法, 并延用至今.

但是, 豫解核求解法耗时、费力又易出错, 更重要的是利用豫解核方法求解积分方程过程中常常出现大计算量的十分复杂且精确的代数与微分、积分等符号计算问题, 所以, 借助计算机的大容量、高速度的特点, 用精确的符号计算, 机械化地实现积分方程求解显得重要而迫切. 基于这种考虑, 本节主要目的是建立基于计算机代数系统Maple的求解第二种 Fredholm 积分方程的机械化算法, 并通过具体算例展示机械化求解积分方程的优势.

2.1.1　豫解核求解法

首先, 我们重述豫解核求解法, 由于该方法对建立机械化算法非常重要, 因此将详述该算法.

考虑如下形式的第二种 Fredholm 积分方程:

$$x(s) = \lambda \int_a^b k(s,t)\, x(t)dt + y(s), \quad a \leqslant s,\, t \leqslant b. \tag{2.1}$$

其中, $\lambda \neq 0$ 是一个参数 (一般情况下可以是复数), 通常用于描述特定的物理量, 自由项 $y(s) \in L^2[a,b]$ 已知, $k(s,t) \in L^2[a,b] \times [a,b]$ 为积分核, 而 $x(s)$ 是要求解的函数.

定理 2.1: 设 $y(s) \in L^2[a,b]$, $k(s,t) \in L^2[a,b] \times [a,b]$ 是连续函数, 若 $d(\lambda) \neq 0$, 则 (2.1) 存在唯一的连续解:

$$x(s) = y(s) + \lambda \int_a^b k_\lambda(s,t) y(t)\, dt, \quad a \leqslant s, t \leqslant b, \tag{2.2}$$

其中, $k_\lambda(s,t)$ 称为豫解核:

$$k_\lambda(s,t) = \frac{D_\lambda(s,t)}{d(\lambda)}, \tag{2.3}$$

这里, $d(\lambda)$ 称为 $k(s,t)$ 的 Fredholm 行列式 (Fredholm determinant), $D_\lambda(s,t)$ 称为 Fred- holm 子式 (Fredholm minor):

$$d(\lambda) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \lambda^n B_n, \tag{2.4}$$

$$D_\lambda(s,t) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \lambda^n A_n(s,t), \tag{2.5}$$

其中,

$$A_0 = k(s,t),$$

$$A_n = \underbrace{\int_a^b \cdots \int_a^b}_{n \text{ times}} \begin{vmatrix} k(s,t) & \cdots & k(s,s_n) \\ k(s_1,t) & \cdots & k(s_1,s_n) \\ \cdots & \cdots & \cdots \\ k(s_n,t) & \cdots & k(s_n,s_n) \end{vmatrix} ds_1 \cdots ds_n, \tag{2.6}$$

$$B_0 = 1,$$

$$B_n = \underbrace{\int_a^b \cdots \int_a^b}_{n \text{ times}} \begin{vmatrix} k(s_1,s_1) & \cdots & k(s_1,s_n) \\ k(s_2,s_1) & \cdots & k(s_2,s_n) \\ \cdots & \cdots & \cdots \\ k(s_n,s_1) & \cdots & k(s_n,s_n) \end{vmatrix} ds_1 \cdots ds_n \quad (n = 1, 2, \cdots). \tag{2.7}$$

定理 2.1 的证明过程从略, 有兴趣者可参看文献 [212]P.31–35.

豫解核 $k_\lambda(s,t)$ 对于除 $d(\lambda)$ 的零点外的所有值是关于 λ 的一个解析函数. 注意到式 (2.4) 中第一项 (即 $n=0$) 为 1, 故 $d(\lambda)$ 不恒为 0, 又因为其为一整函数, 故其零点为一有限的或可列的集合, 这些零点恰为核 $k(s,t)$ 的特征值.

但对于大多数问题来说, 公式 (2.6) 并不便于应用. 我们可建立下述迭代关系:

$$
\begin{aligned}
&A_n(s,t) = B_n k(s,t) - n\int_a^b k(s,u)A_{n-1}(u,t)du,\\
&B_n = \int_a^b A_{n-1}(t,t)dt.
\end{aligned}
\tag{2.8}
$$

为了更清楚地理解定理 2.1 所描述的算法, 下面给出一个简单的例子, 通过该例, 我们也可以看到应用豫解核方法求解第二种 Fredholm 方程时的计算复杂性.

例 2.1: 解方程:

$$
x(s) = s + \int_0^1 (s+t)\,x(t)\,dt, \quad (0 \leqslant s,\ t \leqslant 1). \tag{2.9}
$$

这里,

$$
y(s) = s, \quad k(s,t) = s+t, \quad a=0, \quad b=1, \quad \lambda = 1.
$$

根据 (2.4), 我们可得 Fredholm 行列式 $d(\lambda)$:

$$
\begin{aligned}
d(\lambda) &= 1 - \int_0^1 k(s,s)\,ds + \tfrac{1}{2!}\int_0^1\int_0^1 \begin{vmatrix} k(s,s) & k(s,t) \\ k(t,s) & k(t,t) \end{vmatrix} dsdt \\
&\quad - \tfrac{1}{3!}\int_0^1\int_0^1\int_0^1 \begin{vmatrix} k(s,s) & k(s,t) & k(s,u) \\ k(t,s) & k(t,t) & k(t,u) \\ k(u,s) & k(u,t) & k(u,u) \end{vmatrix} dsdtdu + \cdots \\
&= 1 - \int_0^1 2s\,ds + \tfrac{1}{2!}\int_0^1\int_0^1 \begin{vmatrix} 2s & s+t \\ s+t & 2t \end{vmatrix} dsdt \\
&\quad - \tfrac{1}{3!}\int_0^1\int_0^1\int_0^1 \begin{vmatrix} 2s & s+t & s+u \\ s+t & 2t & t+u \\ s+u & t+u & 2u \end{vmatrix} dsdtdu + \cdots \\
&= -\tfrac{1}{12}.
\end{aligned}
$$

而由公式 (2.5) 可得 Fredholm 子式 $D_\lambda(s,t)$ 为:

$$D_\lambda(s,t) = k(s,t) - \int_0^1 \begin{vmatrix} k(s,t) & k(s,u) \\ k(u,t) & k(u,u) \end{vmatrix} du + \frac{1}{2!}\int_0^1\int_0^1 \begin{vmatrix} k(s,t) & k(s,u) & k(s,v) \\ k(u,t) & k(u,u) & k(u,v) \\ k(v,t) & k(v,u) & k(v,v) \end{vmatrix} dudv + \cdots$$

$$= s+t - \int_0^1 \begin{vmatrix} s+t & s+u \\ u+t & 2u \end{vmatrix} du + \frac{1}{2!}\int_0^1\int_0^1 \begin{vmatrix} s+t & s+u & s+v \\ t+u & 2u & u+v \\ v+t & u+v & 2v \end{vmatrix} dudv + \cdots$$

$$= \frac{1}{2}s + \frac{1}{2}t + \frac{1}{3} + st.$$

从而, 由公式 (2.2), 我们可得方程 (2.9) 的唯一连续解为:

$$x(s) = s + \int_0^1 \frac{D_\lambda(s,t)}{d(\lambda)} \cdot t\,dt = s + \int_0^1 \frac{t(\frac{1}{2}s + \frac{1}{2}t + \frac{1}{3} + st)}{-\frac{1}{12}} dt = -6s - 4.$$

通过这个例子, 我们可以看到, 即使对于 $k(s,t) = s + t$ 这样简单的情形, 其求解过程依然繁琐且计算量很大, 可以想象对于更复杂的情形, 利用手工求解的复杂度更是无法估量. 所以, 利用计算机的大容量、高速度、可进行精确符号计算的特点, 在计算机代数系统下建立机械化算法, 实现积分方程自动求解显得重要而有意义.

2.1.2 Fredholm 积分方程的机械化算法

本节, 借助计算机代数系统`Maple`建立利用豫解核方法求解第二种 Fredholm 积分方程 (2.1) 的机械化算法`fredholmproc`. 显然, 公式 (2.3)—(2.8) 以及解的表达式 (2.2) 为机械化算法的建立奠定了基础, 这些公式均可以很方便地在`Maple`下进行算法设计和程序设计.

为了进一步理解 Fredholm 积分方程的机械化求解算法设计的思想, 首先给出机械化算法设计的流程图 (参见图 2.1).

当利用算法`fredholmproc` 求解方程 (2.1) 时, 所要做的全部事情就是在`Maple`中输入描述方程的信息, 然后机械化算法`fredholmproc`将给出方程 (2.1) 的解. 另一方面, 考虑到数学推导过程的可读性, 我们利用`lprint` 和`print` 设计了相应的求解过程输出语句. 通过下节的例子, 人们可以看到, 利用`fredholmproc` 求解方程 (2.1) 时所得到的结果与我们用笔和纸求解方程时的过程几乎是一样的.

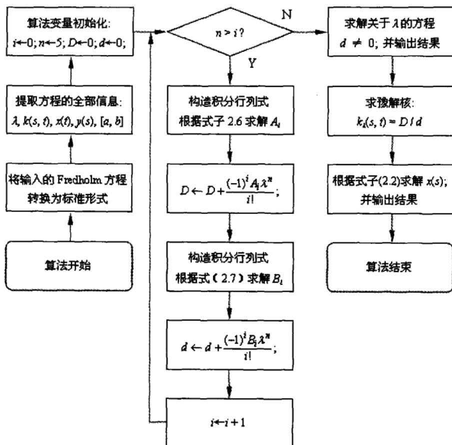


图 2.1 Fredholm 积分方程机械化求解流程图

为了更加清楚地理解积分方程 (2.1) 求解的机械化算法设计过程, 下面, 我们列出算法fredholmproc的源代码:

```
fredholmproc:=proc(expr)
local Expr,parme,i,j,lambd,K,N,xfunction,yfunction,bounds,
      result,d_lamb,D_lamb,Temp1,KdetT,Temp,Xkey,KdetT1;
result:=0;
N:=5;
Expr:=Union(expr);
parme:=parmeint(Expr);
```

```
lambd:=parme[1];
K:=unapply(parme[2],s,t);
xfunction:=parme[3];
yfunction:=parme[4];
bounds:=parme[5];
yfunction:=unapply(yfunction,s);
d_lamb:=1;
for i from 1 to N do
    KdetT:=Kdet(i,K);
    Temp:=KdetT[1];
    for j from 1 to i do
       Temp:=int(Temp,s[j]=bounds[1]..bounds[2]);
    end do;
    d_lamb:=d_lamb+lambd^i*1/i!*(-1)^i*Temp;
    KdetT1:=Kdet(i+1,K);
    Temp1:=KdetT1[2];
    if i=1 then
       D_lamb:=Kdet(2,K)[2];
    else
       for j from 3 to i+1 do
          Temp1:=int(Temp1,s[j]=bounds[1]..bounds[2]);
       end do;
       D_lamb:=D_lamb+lambd^(i-1)*1/(i-1)!*(-1)^(i-1)*Temp1;
    fi;
end do;
KKK:=[solve(d_lamb,lambda)];
D_lamb:=simplify(D_lamb);
lprint(`Here, the parameters of the equation are as:`);
print('y(s)'=yfunction(s),'k(s,t)'=K(s,t));
lprint(`When parameter lambda satisfy with:`);
if not has(d_lamb,lambda) then
    print(lambda=arbitrary constant);
fi;
if has(d_lamb,lambda) then
    for i from 1 to nops(KKK) do
```

```
        print(lambda<>KKK[i]);
      od;
  fi;
  lprint('Here, the parameters of the equation are as:');
  print('y(s)'=yfunction(s),'k(s,t)'=K(s,t));
  lprint('Then,the Fredholm determinant of the kernel is
          as:');
  print('d(lambda)'=d_lamb);
  lprint('And the Fredholm minor is:');
  print('D[lambda](s,t)'=subs(s[1]=s,s[2]=t,D_lamb));
  Xkey:=normal(yfunction(s[1])+lambd*int(D_lamb/d_lamb
          *yfunction(s[2]),
        s[2]=bounds[1]..bounds[2]));
  lprint('The solution of the equation is as follows:');
  print('x(s)'=subs(s[1]=s,Xkey));
  end proc:
```

其中, 参数expr表示要求解的第二种 Fredholm 积分方程的信息. 例如, 对于积分方程 (2.1) 而言, 我们在Maple下只需输入下述命令即可:

```
expr:=x(s)=lambda*int(k(s,t)*x(t),t=a..b)+y(s);
fredholmproc(expr);
```

然后运行fredholmproc, 我们就可以得到方程的解.

另外, 在主程序fredholmproc中有两个子程序Union 和parmeint. 算法Union 是将方程 (2.1) 转换为:

$$x(s)-\lambda\int_a^b k(s,t)x(t)dt-y(s)=0 \tag{2.10}$$

的标准形式. 事实上, 积分方程求解机械化算法设计针对的是一般形式的或者称之为标准形式的方程, 算法Union 所做的正是将输入的方程转化为标准形式, 即方程 (2.10) 之形式. 而对于刻划方程的信息的提取是机械化求解中至关重要的一个步骤, 通过算法parmeint 可以看出, 信息的提取其实是将方程的各个部分分解出来. 对于积分方程而言, 这些部分包括参数 λ, 核函数 $k(s,t)$, 函数 $y(s)$, 积分限 $[0,s]$ 以及待求解函数 $x(s)$. 算法parmeint的主要功能就是提取方程 (2.10) 中的上述信息.

作为例子, 我们将算法parmeint的源代码列示如下:

```
parmeint:=proc(expr)
local hexpr,temp,i,k,xfunction,yfunction,lambda,TEMP,
          bounds,x_int;
hexpr[1]:=lhs(expr);
hexpr[2]:=rhs(expr);
temp:=[ ];
TEMP:=[ ];
yfunction:=0;
for i from 1 to 2 do
    if hexpr[i]<>0 then
       if type(hexpr[i],'+') and epxr[i]<>0 then
          temp:=[op(temp),op(hexpr[i])];
       else
          temp:=[op(temp),hexpr[i]];
       fi;
    fi;
od;
for i from 1 to nops(temp) do
    if has(temp[i],'int') then
       k:=temp[i];
    elif has(temp[i],x) then
       xfunction:=temp[i];
    else
       yfunction:=yfunction+temp[i]
    fi;
od;
if type(k,'*') then
    lambda:=convert([seq(op(i,k),i=1..nops(k)-1)],'*');
    k:=op(nops(k),k);
else
    lambda:=1;  k:=k;
fi;
temp:=[op(op(1,k))];
bounds:=[op(op(2,op(2,k)))];
for i from 1 to nops(temp) do
```

```
        if has(temp[i],x) then
            x_int:=temp[i];
        else
            TEMP:=[op(TEMP),temp[i]];
        fi;
    od;
    k:=convert(TEMP,'*');
    return (-lambda, k, xfunction, -yfunction, bounds);
    end:
```

算法parmeint的设计思想在积分方程求解机械化研究中具有普适性, 不仅适用于 Fredholm 积分方程, 也适用于第三章要研究的 Volterra 积分方程.

另外, 需要指出的是, 本书所有程序均在 IBM-R50e(奔腾 1.60GHz, 512M 内存) 笔记本电脑中运行通过, 所用Maple为 Version 11.0.

2.1.3 应用算例

本节, 我们将通过下述几个例子展示算法fredholmproc的高效性与快捷性.

例 2.2: 首先考虑如下 Fredholm 积分方程的求解:

$$x(s) = s + \lambda \int_0^1 (4\,st - s^2)\, x(t)\, dt. \tag{2.11}$$

在Maple下, 我们需要输入下述命令:

```
expr1:=x(s)=s+lambda*int((4*s*t-s^2)*x(t),t=0..1);
fredholmproc(expr1);
```

然后, 运行机械化算法fredholmproc即可得到下述求解过程:

`Here, the parameters of the equation are as:`

$$y(s) = s,\ \ k(s,t) = 4st - s^2$$

`When parameter lambda satisfy with:`

$$\lambda \neq 9 + 3\sqrt{7}, \quad \lambda \neq 9 - 3\sqrt{7}$$

`The Fredholm determinant of the kernel is as:`

$$d\,(\lambda) = 1 - \lambda + \frac{1}{18}\,\lambda^2$$

`and the Fredholm minor is:`

$$D_\lambda(s,t) = 4\,st - s^2 - \lambda\,s + \frac{4}{3}\,\lambda\,st + \frac{4}{3}\,\lambda\,s^2 - 2\,\lambda\,s^2 t$$

`So, we can get the solution of the equation as:`

$$x(s) = -\frac{3s\,(-2\,\lambda - 6 + 3\,\lambda\,s)}{18 - 18\,\lambda + \lambda^2}$$

在`Maple`下, 利用计时命令`time`可以获知求解例 2.2 的时间仅为 0.201 秒. 由此, 对机械化算法`fredholmproc`求解积分方程的快捷性可窥一斑.

另外, 需要说明的是, 从本例可以看到, 机械化求解积分方程的过程即我们所设计的可读性的输出语句全部用英语表述, 主要原因是在`Maple`下虽然可以输入汉语, 但程序运行时有时会出现乱码. 这一求解过程的英语表述习惯贯穿全文.

例 2.3: 求解积分核为有理函数的方程:

$$x(s) = s^2 + \int_0^1 \frac{1+s+s^2}{2+t}\,x(t)\,dt. \tag{2.12}$$

输入命令:

```
expr2:=x(s)=s^2+int((1+s+s^2)/(2+t)*x(t),t=0..1);
fredholmproc(expr2);
```

通过机械化算法可得:

`Here, the parameters of the equation are as:`

$$y(s) = s^2,\ k(s,t) = \frac{1+s+s^2}{2+t}$$

`When parameter lambda satisfy with:`

$$\lambda = \texttt{arbitrary number}$$

`The Fredholm determinant of the kernel is as:`

$$d\,(\lambda) = \frac{3}{2} - 3\,\ln(3) + 3\,\ln(2)$$

`and the Fredholm minor is:`

$$D_\lambda(s,t) = \frac{1+s+s^2}{2+t}$$

So, we can get the solution of the equation as:

$$x(s) = -\frac{1}{3}\,\frac{-2\,s^2\ln(3)+2\,s^2\ln(2)-8\,\ln(3)+8\,\ln(2)+3-8\,s\ln(3)+8\,s\ln(2)+3\,s}{1-2\,\ln(3)+2\,\ln(2)}$$

机械化算法给出的上式求解结果显然不是最简形式，通过化简我们容易得到:

$$x(s) = \frac{1}{3}\,\frac{2(\ln 3-\ln 2)(s+2)^2}{1-2(\ln 3-\ln 2)}$$

事实上，这是计算机代数系统Maple下的一个习惯，即从不自动化简. 这样做有时候会有一些好处 (譬如，不会因为化简而丢失信息)，但有时候会有一些不便. 但无论怎么说，都并不影响结果的正确性.

例 2.4: 考虑 Fredholm 积分方程:

$$x(s) = \lambda\int_0^1 \sin(\ln s)\,x(t)\,dt + 2\,s. \tag{2.13}$$

在Maple中输入:

```
expr3:=x(s)=lambda*int(sin(ln(s))*x(t),t=0..1)+2*s;
fredholmproc(expr3);
```

可得机械化求解过程如下:

Here, the parameters of the equation are as:

$$y(s) = 2s, \quad k(s,t) = \sin(\ln(s))$$

When parameter lambda satisfy with:

$$\lambda \neq -2$$

The Fredholm determinant of the kernel is as:

$$d(\lambda) = 1+\frac{\lambda}{2}$$

and the Fredholm minor is:

$$D_\lambda(s,t) = \sin(\ln(s))$$

So, we can get the solution of the equation as:

$$x(s) = 2\,\frac{2\,s + s\lambda + \lambda\,\sin(\ln(s))}{2+\lambda}$$

例 2.5: 解方程:

$$x(s) = s + \lambda \int_{-\pi}^{\pi} \left(s\cos t + t^2 \sin s + \cos s \sin t\right)\, x(t)\, dt. \tag{2.14}$$

在Maple下输入命令:

```
expr4:=x(s)=s+lambda*int((s*cos(t)+t^2*sin(s)+cos(s)
      *sin(t))*x(t),t=-Pi..Pi);
fredholmproc(expr4);
```

则机械化算法给出如下求解过程:

Here, the parameters of the equation are as:

$$y(s) = s,\;\; k(s,t) = s\cos(t) + t^2\sin(s) + \cos(s)\sin(t)$$

When parameter lambda satisfy with:

$$\lambda \neq \frac{\frac{1}{2}\,i\sqrt{2}}{\pi},\quad \lambda \neq \frac{-\frac{1}{2}\,i\sqrt{2}}{\pi}$$

The Fredholm determinant of the kernel is as:

$$d(\lambda) = 1 + 2\,\lambda^2\pi^2$$

and the Fredholm minor is:

$$\begin{aligned} D_\lambda(s,t) = {} & s\cos(t) + t^2\sin(s) + \cos(s)\sin(t) + \lambda\, s\sin(t)\,\pi - 4\,\lambda\,\sin(s)\sin(t)\,\pi \\ & +2\,\lambda\,\cos(s)\cos(t)\,\pi + \lambda\,\cos(s)\,t^2\pi - 2\,\lambda^2 t^2\sin(s)\,\pi^2 + 4\,\lambda^2 s\cos(t)\,\pi^2 \\ & -8\,\lambda^2\cos(t)\sin(s)\,\pi^2 + \lambda^2 s t^2\pi^2 \end{aligned}$$

So, we can get the solution of the equation as:

$$x(s) = \frac{s + 2\,\lambda\,\cos(s)\,\pi + 4\,\lambda^2 s\pi^2 - 8\,\lambda^2\sin(s)\,\pi^2}{1 + 2\,\lambda^2\pi^2}$$

例 2.6: 求解:

$$x(s) = s + \lambda \int_0^{\pi} (1 + \sin s \sin t)\ x(t)\, dt. \tag{2.15}$$

在Maple下输入命令:

```
expr5:=x(s)=s+lambda*int((1+sin(s)*sin(t))*x(t),t=0..Pi);
fredholmproc(expr5);
```

则会得到下述求解过程:

```
Here, the parameters of the equation are as:
```

$$y(s) = s,\ k(s,t) = 1 + \sin(s)\sin(t)$$

```
When parameter lambda satisfy with:
```

$$\lambda \neq \frac{1}{2}\,\frac{3\,\pi + \sqrt{\pi^2+64}}{\pi^2-8},\quad \lambda \neq \frac{1}{2}\,\frac{3\,\pi - \sqrt{\pi^2+64}}{\pi^2-8}$$

```
The Fredholm determinant of the kernel is as:
```

$$d\,(\lambda) = 1 - \frac{3}{2}\,\lambda\,\pi + \frac{1}{2}\,\lambda^2\left(\pi^2-8\right)$$

```
and the Fredholm minor is:
```

$$D_\lambda(s,t) = 1 + \sin(s)\sin(t) - \frac{1}{2}\,\lambda\,\pi - \lambda\,\sin(s)\sin(t)\,\pi + 2\,\lambda\,\sin(t) + 2\,\lambda\,\sin(s)$$

```
So, we can get the solution of the equation as:
```

$$x\,(s) = -\frac{1}{2}\,\frac{-4\,s + 6\,s\lambda\,\pi - 2\,\lambda^2 s\pi^2 + 16\,\lambda^2 s - 2\,\lambda\,\pi^2 - 4\,\lambda\,\sin(s)\,\pi + \lambda^2\pi^3 - 8\,\lambda^2\pi}{2 - 3\,\lambda\,\pi + \lambda^2\pi^2 - 8\,\lambda^2}$$

上式给出的方程 (2.15) 的解不是最简形式, 我们可以进一步化简为:

$$x\,(s) = s + \frac{\pi\,\lambda\,\left(2\,\pi - \left(\pi^2-8\right)\lambda\right) + 4\,\sin(s)}{4 - 6\,\lambda\,\pi + 2\,\lambda^2\left(\pi^2-8\right)}$$

例 2.7: 求解:

$$x(s) - \lambda \int_0^1 \arccos t\, x(t)\, dt = \frac{1}{\sqrt{1-s^2}}. \tag{2.16}$$

在Maple中, 输入命令:

```
expr6:=x(s)-lambda*int(arccos(t)*x(t),t=0..1)=1/sqrt(1-s^2);
fredholmproc(expr6);
```

即得求解过程:

Here, the parameters of the equation are as:

$$y(s) = \frac{1}{\sqrt{1-s^2}},\ k(s,t) = \arccos(t)$$

When parameter lambda satisfy with:

$$\lambda \neq 1$$

The Fredholm determinant of the kernel is as:

$$d(\lambda) = 1 - \lambda$$

and the Fredholm minor is:

$$D_\lambda(s,t) = \arccos(t)$$

So, we can get the solution of the equation as:

$$x(s) = -\frac{1}{8}\,\frac{8 - 8\lambda + \lambda\pi^2\sqrt{1-s^2}}{\sqrt{1-s^2}\,(-1+\lambda)}$$

例 2.8: 解方程:

$$x(s) - \lambda\int_0^1 (s\ln t - t\ln s)\,x(t)\,dt = \frac{6}{5}\Big(1 - 4s\Big). \tag{2.17}$$

输入命令:

```
expr7:=x(s)-lambda*int((s*ln(t)-t*ln(s))*x(t),t=0..1)
        =6/5*(1-4*s);
fredholmproc(expr7);
```

则机械化求解过程为:

```
Here, the parameters of the equation are as:
```

$$y(s)=\frac{6}{5}-\frac{24}{5}s,\ k(s,t)=s\ln(t)-t\ln(s)$$

```
When parameter lambda satisfy with:
```

$$\lambda\neq\frac{4}{29}\sqrt{87}\,i,\quad \lambda\neq\frac{-4}{29}\sqrt{87}\,i$$

```
The Fredholm determinant of the kernel is as:
```

$$d(\lambda)=1+\frac{29}{48}\lambda^2$$

```
and the Fredholm minor is:
```

$$D_\lambda(s,t)=s\ln(t)-t\ln(s)-\frac{1}{3}\lambda\ln(s)\ln(t)-\frac{1}{4}\lambda s\ln(t)-\frac{1}{4}\lambda t\ln(s)-2\lambda st$$

```
So, we can get the solution of the equation as:
```

$$x(s)=\frac{6}{5}\,\frac{48+29\lambda^2-192s-36\lambda^2s+40\lambda\ln(s)+10\lambda^2\ln(s)}{48+29\lambda^2}$$

例 2.9: 求解方程:

$$x(s)-\lambda\int_0^{2\pi}|\pi-t|\sin s\,x(t)\,dt=s. \tag{2.18}$$

在Maple中输入:

```
expr8:=x(s)-lambda*int(abs(Pi-t)*sin(s)*x(t),t=0..2*Pi)=s;
fredholmproc(expr8);
```

即得求解过程为:

```
Here, the parameters of the equation are as:
```

$$y(s)=s,\ k(s,t)=|\pi-t|\sin(s)$$

```
When parameter lambda satisfy with:
```

$$\lambda=\texttt{arbitrary number}$$

```
The Fredholm determinant of the kernel is as:
```

$$d(\lambda)=1$$

and the Fredholm minor is:

$$D_\lambda(s,t) = |-\pi + t| \sin(s)$$

So, we can get the solution of the equation as:

$$x(s) = s + \lambda \pi^3 \sin(s)$$

例 2.10: 解方程:

$$x(s) - \lambda \int_0^\pi \frac{s-t}{\sqrt{\pi+t}} x(t)\, dt = s. \tag{2.19}$$

在Maple中输入命令:

```
expr9:=x(s)-lambda*int((Pi+t)^(-1/2)*(s-t)*x(t),t=0..Pi)
        =s;
fredholmproc(expr9);
```

即得求解过程为:

Here, the parameters of the equation are as:

$$y(s) = s,\ k(s,t) = \frac{s-t}{\sqrt{\pi+t}}$$

When parameter lambda satisfy with:

$$\lambda \neq \frac{6\,i\sqrt{180\,\pi^3 - 125\,\pi^3\sqrt{2}}}{144\,\pi^3 - 100\,\pi^3\sqrt{2}},\quad \lambda \neq \frac{-6\,i\sqrt{180\,\pi^3 - 125\,\pi^3\sqrt{2}}}{144\,\pi^3 - 100\,\pi^3\sqrt{2}}$$

The Fredholm determinant of the kernel is as:

$$d(\lambda) = 1 + \frac{1}{2}\lambda^2\left(\frac{32}{5}\pi^3 - \frac{40}{9}\pi^3\sqrt{2}\right)$$

and the Fredholm minor is:

$$D_\lambda(s,t) = -\frac{1}{15\sqrt{\pi+t}}\Big(-15\,s + 15\,t + 14\sqrt{2}\,\lambda\,\pi^{\frac{5}{2}} + 10\sqrt{2}\,\lambda\,\pi^{\frac{3}{2}}s + 10\sqrt{2}\,\lambda\,\pi^{\frac{3}{2}}t$$
$$+30\sqrt{2}\,\lambda\sqrt{\pi}st - 16\,\lambda\,\pi^{\frac{5}{2}} - 20\,\lambda\,\pi^{\frac{3}{2}}s - 20\,\lambda\,\pi^{\frac{3}{2}}t - 30\sqrt{\pi}\,\lambda\, st\Big)$$

So, we can get the solution of the equation as:

$$x(s) = \frac{3\left(-15\,s + 14\sqrt{2}\,\lambda\,\pi^{\frac{5}{2}} + 10\sqrt{2}\,\lambda\,\pi^{\frac{3}{2}}s - 16\,\lambda\,\pi^{\frac{5}{2}} - 20\,\lambda\,\pi^{\frac{3}{2}}s\right)}{-45 - 144\,\lambda^2\pi^3 + 100\sqrt{2}\,\lambda^2\pi^3}$$

本节最后，我们列示一个 $d(\lambda)$ 存在 2 阶零点的例子.

例 2.11: 求解方程:

$$x(s) = 1 + s + s^2 + \lambda \int_{-1}^{1} (1 + t + 3st)\, x(t)\, dt. \tag{2.20}$$

在Maple中输入命令:

```
expr10:=x(s)=1+s+s^2+lambda*int((1+t+3*s*t)*x(t),t=-1..1);
fredholmproc(expr10);
```

即可得:

Here, the parameters of the equation are as:

$$y(s) = 1 + s + s^2,\ \ k(s,t) = 1 + t + 3st$$

When parameter lambda satisfy with:

$$\lambda \neq \frac{1}{2}$$

The Fredholm determinant of the kernel is as:

$$d(\lambda) = 1 - 4\lambda + 4\lambda^2$$

and the Fredholm minor is:

$$D_\lambda(s,t) = 1 + t + 3st - 2\lambda - 6\lambda st$$

So, we can get the solution of the equation as:

$$x(s) = \frac{3 - 2\,\lambda - 4\,\lambda^2 + 3\,s - 6\,\lambda\,s + 3\,s^2 - 12\,s^2\lambda + 12\,s^2\lambda^2}{3(1 - 4\,\lambda + 4\,\lambda^2)}$$

从以上算例的结果可以看出，本节所建立的求解第二种 Fredholm 积分方程的机械化算法fredholmproc可以给出方程 (2.1) 的唯一连续解的解析式，而且也能给出解方程的过程. 应用算例结果表明，该算法对于求解 Ferdholm 积分方程 (2.1) 简单实用且高效快捷. 所得结果有时候并非我们所想象的最简单的形式，但这并不影响结果的正确性.

2.2　Fredholm 积分方程组求解

上节讨论了 Fredholm 积分方程 (2.1) 求解问题并在计算机代数系统Maple 下建立了机械化算法fredholmproc,但在实际问题中常常会遇到如下形式的 Fredholm 积分方程组求解问题:

$$x_i(s) - \lambda \sum_{j=1}^{n} \int_a^b k_{ij}(s,t)x_j(t)dt = y_i(s), \quad i = 1,2,\cdots,n. \tag{2.21}$$

其中, $k_{ij}(s,t)$ 和 $y_j(s)$ $(i,j=1,2,\cdots,n)$ 都是平方绝对可积的已知函数, 即:

$$\int_a^b \int_a^b |k_{ij}(s,t)|^2 dsdt < \infty, \quad \int_a^b |y_j(t)|^2 dt < \infty \quad (i,j=1,\cdots,n).$$

而 $x_i(s)\,(i=1,2,\cdots,n)$ 是未知函数.

2.2.1　基本算法

Fredholm 积分方程组 (2.21) 的求解方法与单个方程 (2.1) 基本上类似. 当 λ 满足

$$|\lambda| < \left\{ \max_{1\leqslant i\leqslant n} \sqrt{\int_a^b \int_a^b |k_{ij}(s,t)|^2 dsdt} \right\}^{-1}$$

时, 应用逐次逼近法所得的迭代序列平均收敛于方程组 (2.21) 的解. 而当 $k_{ij}(s,t)$ 是一退化核时, 方程组 (2.21) 可化为一代数方程组求解.

下面, 给出将 Fredholm 积分方程组 (2.21) 化为单个 Fredholm 积分方程的方法. 考虑 s 和 t 变化的范围是 $[a, nb-(n-1)a]$, 并定义函数 $K(s,t)$, $Y(s)$, $X(s)$:

当 $(i-1)b-(i-2)a \leqslant s \leqslant ib-(i-1)a$ 时, 令

$$X(s) = x_i(s-(i-1)(b-a)), \quad Y(s) = y_i(s-(i-1)(b-a)); \tag{2.22}$$

当 $(i-1)b-(i-2)a \leqslant s \leqslant ib-(i-1)a$, $(m-1)b-(m-2)a \leqslant t \leqslant mb-(m-1)a$ 时, 令

$$K(s,t) = k_{im}(s-(i-1)(b-a), t-(m-1)(b-a)). \tag{2.23}$$

这里, $i,m = 1,2,\cdots,n$.

由此, 方程组 (2.21) 可化为与方程 (2.1) 形式上完全一样的方程:

$$X(s) - \lambda \int_a^{nb-(n-1)a} K(s,t)X(t)dt = Y(s), \quad i = 1,2,\cdots,n. \tag{2.24}$$

其中,

$$X(s) = (x_1(s), x_2(s-(b-a)), \cdots, x_n(s-(n-1)(b-a)))^T,$$

$$Y(s) = (y_1(s), y_2(s-(b-a)), \cdots, y_n(s-(n-1)(b-a)))^T,$$

$$K(s,t) = \begin{pmatrix} k_{11}(s,t) & \cdots & k_{1n}(s,t-(n-1)(b-a)) \\ k_{21}(s-(b-a),t) & \cdots & k_{2n}(s-(b-a),t-(n-1)(b-a)) \\ \cdots & \cdots & \cdots \\ k_{n1}(s-(n-1)(b-a),t) & \cdots & k_{nn}(s-(n-1)(b-a),t-(n-1)(b-a)) \end{pmatrix}.$$

矩阵 $K(s,t)$ 称为向量积分方程 (2.24) 的核. 这种向量形式的方程和通常一个未知函数的积分方程 (2.1) 的相似之处不仅仅是表面的, 而且关于豫解核求解法 (定理 2.1) 几乎不需要作任何改变就可移植到由方程组 (2.21) 改写而得到的向量方程 (2.24) 上. 根据定理 2.1, 只需要当 $y_i(s) \in L^2[a,b]$, $k_{ij}(s,t) \in L^2[a,b] \times [a,b]$ 时, 按 (2.22) 定义的 $Y(s)$ 及按 (2.23) 定义的 $K(s,t)$ 分别是区间 $[a, nb-(n-1)a]$ 和正方形 $[a, nb-(n-1)a] \times [a, nb-(n-1)a]$ 上的 L^2- 和 L^2- 核函数即可. 这一点是容易满足的. 事实上, 我们有:

$$\begin{aligned}
&\int_a^{nb-(n-1)a}\int_a^{nb-(n-1)a} |K(s,t)|^2 ds dt \\
&= \sum_{i,k=1}^{n} \int_{(m-1)b-(m-2)a}^{mb-(m-1)a} \int_{(i-1)b-(i-2)a}^{ib-(i-1)a} |K(s,t)|^2 ds dt \\
&= \sum_{i,m=1}^{n} \int_a^b \int_a^b |K(s+(m-1)(b-1), t+(i-1)(b-a))|^2 ds dt \\
&= \sum_{i,m=1}^{n} \int_a^b \int_a^b |k_{im}(s,t)|^2 ds dt < \infty.
\end{aligned}$$

同理可证

$$\int_a^{nb-(n-1)a} |Y(s)|^2 ds = \sum_{i=1}^{n} \int_a^b |y_i(s)|^2 ds < \infty.$$

为了更清楚地描述利用与定理 2.1 几乎完全类似的豫解核方法求解 Fredholm 积分方程组 (2.21) 或 (2.24), 下面给出一个简单的算例.

例 2.12: 求解 Fredholm 方程组:

$$x_1(s) - \int_0^1 t x_2(s) dt = -\frac{s}{2}, \quad x_2(s) - \int_0^1 t x_1(s) dt = s + \frac{3}{2}. \tag{2.25}$$

注意到, 这里, $\lambda = 1, a = 0, b = 1, n = 2$, 把方程组 (2.25) 改写成向量方程 (2.24) 的形式:

$$X(s) - \int_0^2 K(s,t)X(t)dt = Y(s), \tag{2.26}$$

其中,

$$X(s) = (x_1(s), x_2(s-1))^T, \quad Y(s) = (y_1(s),\, y_2(s-1))^T = (-\tfrac{s}{2},\, s-\tfrac{1}{3})^T,$$

$$K(s,t) = \begin{pmatrix} k_{11}(s,t) & k_{12}(s,t-1) \\ k_{21}(s-1,t) & k_{22}(s-1,t-1) \end{pmatrix} = \begin{pmatrix} 0 & s \\ t & 0 \end{pmatrix}.$$

根据上节的公式 (2.7) 可计算得:

$$B_0 = 1, \quad B_1 = \textstyle\int_0^2 K(s,s)ds = \int_0^1 k_{11}(s,s)ds + \int_1^2 k_{12}(s,s-1)ds = 0;$$

$$\begin{aligned} B_2 &= \int_0^2\int_0^2 \begin{vmatrix} k_{11}(s,s) & k_{12}(s,t-1) \\ k_{21}(t-1,s) & k_{22}(t-1,t-1) \end{vmatrix} dsdt \\ &= \int_0^1\int_0^1 \begin{vmatrix} k_{11}(s,s) & k_{12}(s,t-1) \\ k_{21}(t-1,s) & k_{22}(t-1,t-1) \end{vmatrix} dsdt \\ &+ \int_0^1\int_1^2 \begin{vmatrix} k_{11}(s,s) & k_{12}(s,t-1) \\ k_{21}(t-1,s) & k_{22}(t-1,t-1) \end{vmatrix} dsdt \\ &+ \int_1^2\int_0^1 \begin{vmatrix} k_{11}(s,s) & k_{12}(s,t-1) \\ k_{21}(t-1,s) & k_{22}(t-1,t-1) \end{vmatrix} dsdt \\ &+ \int_1^2\int_1^2 \begin{vmatrix} k_{11}(s,s) & k_{12}(s,t-1) \\ k_{21}(t-1,s) & k_{22}(t-1,t-1) \end{vmatrix} dsdt \\ &= -\tfrac{2}{3}, \end{aligned}$$

$$B_3 = B_4 = \cdots = 0,$$

从而, 由公式 (2.4) 可得 Fredholm 行列式为:

$$d(\lambda) = \sum_{i=0}^{\infty} \frac{(-1)^i}{i!}\lambda^i B_i = B_0 - B_1 + \frac{1}{2!}B_2 = 1 + 0 - \frac{1}{3} = \frac{2}{3}.$$

同样, 由公式 (2.5) 可计算得到第一阶 Fredholm 子式为:

$$D_\lambda(s,t) = \frac{1}{3}.$$

则由公式 (2.2) 可得方程 (2.26) 的解为:

$$X(s) = Y(s) + \int_0^2 \frac{D_\lambda(s,t)}{d(\lambda)} Y(s)dt = Y(s) + 2\int_0^2 Y(s)dt,$$

当 $0 \leqslant s \leqslant 1$ 时,

$$x_1(s) = -\frac{s}{2} + 2\int_0^1 (-\frac{s}{2})dt + 2\int_1^2 (s - \frac{1}{3})dt = s,$$

当 $1 \leqslant s \leqslant 2$ 时,

$$x_2(s) = s - \frac{1}{3} + 2\int_0^1 (-\frac{s}{2})dt + 2\int_1^2 (s - \frac{1}{3})dt = s + 1,$$

综上, 即求得 Fredholm 方程组 (2.25) 的解为:

$$x_1(s) = s, \quad x_2(s) = s + 1.$$

由例 2.12 的计算过程可以看出, Fredholm 积分方程组求解过程较单个 Fredholm 方程的情形计算更为复杂, 下面, 我们将利用豫解核求解法在Maple下建立相应的机械化算法.

2.2.2 Fredholm 积分方程组的机械化算法

从上节的基本算法及例 2.12 的计算过程可以看出, 当 Fredholm 积分方程组 (2.21) 转换为向量积分方程 (2.24) 后, 其求解方法和单个 Fredholm 方程 (2.1) 求解方法极其类似, 为了避免重复, 本节只列示求解 (2.21) 的主程序, 其余相关子程序可参看上节相关内容.

```
FredEqns:=proc(expr)
local Expr, parme, i, N, result, num, lambd, TKList, K,
    xfunction, yfunction, bounds, d_lamb, D_lamb, m, n,
    inter, phi, j, kk, Temp;
result:=0;
num:= nops(expr);
Expr:=map(_u->Union(_u),expr);
for i from 1 to num do
parme[i]:=parmeint(Expr[i],2);
end do:
lambd:=parme[1][1];
TKList:=[seq(map(unapply,parme[i][2],s,t),i=1..num)];
```

```
K:=Matrix(num,2,TKList);
bounds:=parme[1][5];
inter:=bounds[2]-bounds[1];
xfunction:=map(unapply,[seq(parme[i][3],i=1..num)],s);
yfunction:=map(unapply,[seq(parme[i][4],i=1..num)],s);
for m from 1 to num do
    for n from 1 to num do
       K[m,n]:=K[m,n](s-(m-1)*(bounds[2]-bounds[1]),t-(n-1)
          *(bounds[2]-bounds[1]));
       K[m,n]:=unapply(K[m,n],s,t);
    end do:
end do:
d_lamb:=1;  D_lamb:=0;  phi:=[seq(0,i=1..num)];
for i from 1 to num do
    for j from 1 to num do
       d_lamb:=1;
       D_lamb:=0;
       for kk from 1 to N do
         Temp:=KDetInt(num,kk,K,bounds,i,j);
         d_lamb:=d_lamb+lambda^kk*Temp[1];
         D_lamb:=D_lamb+lambda^kk*Temp[2];
       end do:
       phi[i]:=phi[i]+int(D_lamb/d_lamb*yfunction[j](s[2]
           -(j-1)*inter), s[2]=bounds[1]+(j-1)*inter
           ..bounds[2]+(j-1)*inter);
    end do:
    phi[i]:=phi[i]+yfunction[i](s[1]-(i-1)*inter);
    phi[i]:=unapply(phi[i],s[1]);
    phi[i]:=simplify(phi[i](s+(i-1)*inter));
end do:
xfunction:=seq(xfunction[i](s-(i-1)*inter),i=1..num);
yfunction:=seq(yfunction[i](s-(i-1)*inter),i=1..num);
lprint(`The system can be rewritten as`);
print(X(s)-lambda*int('K(x,s)'*X(t),t=bounds[1]
       ..bounds[2]+(num-1)*inter)=Y(s));
```

```
lprint('where');
print('X(s)'=xfunction,'Y(s)'=yfunction,'K(s,t)'=map
        (_u->_u(s,t),K));
lprint('The Fredholm determinant of the kernel K(s,t) is
      as:');
print('d(lambda)'=d_lamb);
lprint('and the Fredholm minor is:');
print('D[lambda](s,t)'=subs(s[1]=s,s[2]=t,D_lamb));
lprint('So, we can get the solution of the equations
      as:');
print((seq(parme[i][3],i=1..num))=seq(phi[i],i=1..num));
```

2.2.3 应用算例

本小节, 将通过若干算例演示机械化算法FredEqns的使用方法及功能.

例 2.13: 求解 Fredholm 积分方程组:

$$\begin{cases} x_1(s) - \int_0^1 (s+t)\, x_1(t)\, dt - \int_0^1 (s+2t)\, x_2(t)\, dt = -\frac{3}{2}s - \frac{8}{3}, \\ x_2(s) - \int_0^1 (2s+t)\, x_1(t)\, dt - \int_0^1 (2s+2t)\, x_2(t)\, dt = -3s - \frac{5}{3}. \end{cases} \tag{2.27}$$

在Maple中输入下述命令:

```
exprs[1]:=x[1](s)-int((s+t)*x[1](t),t=0..1)-int((s+2*t)
      *x[2](t),t=0..1)=-3/2*s-8/3 ;
exprs[2]:=x[2](s)-int((2*s+t)*x[1](t),t=0..1)-int((2*s
      +2*t)*x[2](t),t=0..1)=-3*s-5/3;
FredEqns([exprs[1],exprs[2]]);
```

即可得到机械化求解结果:

```
The system can be rewritten as:
```

$$X(s) - \lambda \int_0^2 K(s,t) X(s) dt = Y(s)$$

```
where
```

$$X(s) = (x_1(s), x_2(s-1)),$$

$$Y(s) = (-\tfrac{3}{2}s - \tfrac{8}{3}, -3s + \tfrac{4}{3}),$$

$$K(s,t) = \begin{bmatrix} s+t & s+2t-2 \\ 2s-2+t & 2s-4+2t \end{bmatrix}$$

```
The Fredholm determinant of the kernel K(s,t) is as:
```

$$d(\lambda) = -\frac{37}{12}$$

```
and the Fredholm minor is:
```

$$D_\lambda(s,t) = -9s + \frac{35}{3} - 9t + 8st$$

```
So, we can get the solution of the equations as:
```

$$(x_1(s), x_2(s)) = (s, 1+2s)$$

例 2.14: 求解积分方程组:

$$\begin{cases} x_1(s) - \int_0^1 (s+t)\, x_1(t)\, dt - \int_0^1 s x_2(t)\, dt = s^2 - \frac{1}{4} - \frac{7}{3}s, \\ x_2(s) - \int_0^1 (s^2+t)\, x_1(t)\, dt - \int_0^1 (2s+2t)\, x_2(t)\, dt = -2s - \frac{19}{12} - \frac{1}{3}s^2. \end{cases} \tag{2.28}$$

在Maple中输入下述命令:

```
exprs[3]:=x[1](s)-int((s+t)*x[1](t),t=0..1)-int((s)
        *x[2](t),t=0..1)=s^2-1/4-7/3*s;
exprs[4]:=x[2](s)-int((s2+t)*x[1](t),t=0..1)-int((2*s
        +2*t)*x[2](t),t=0..1)=-2*s-19/12-1/3*s^2;
FredEqns([exprs[3],exprs[4]]);
```

即可得到机械化求解结果:

```
The system can be rewritten as:
```

$$X(s) - \lambda \int_0^2 K(s,t) X(s) dt = Y(s)$$

where

$$X(s) = (x_1(s), x_2(s-1)),$$

$$Y(s) = (s^2 - \tfrac{1}{4} - \tfrac{7}{3}\, s, -2\, s + \tfrac{5}{12} - \tfrac{1}{3}\, (s-1)^2),$$

$$K(s,t) = \begin{bmatrix} s+t & s \\ (s-1)^2+t & 2s-4+2t \end{bmatrix}$$

The Fredholm determinant of the kernel K(s,t) is as:

$$d(\lambda) = -\frac{37}{72}$$

and the Fredholm minor is:

$$D_\lambda(s,t) = \frac{8}{3}\, s - \frac{43}{72} + \frac{10}{9}\, t - \frac{7}{6}\, s^2 - \frac{8}{3}\, st + \frac{7}{6}\, ts^2$$

So, we can get the solution of the equation as:

$$(x_1(s), x_2(s)) = (s^2, 1+2s)$$

例 2.15: 求解:

$$\begin{cases} x_1(s) - \int_0^1 (s+t)\, x_1(t)\, dt - \int_0^1 s x_2(t)\, dt = \sin(s) + \frac{1}{2} - 2\, s + \cos(1)\, s \\ \qquad - \sin(1) + \cos(1) - 2\, s\, (\sin(1) + 4)\,, \\ x_2(s) - \int_0^1 \left(s^2 + t\right) x_1(t)\, dt - \int_0^1 (2\, s + 2\, t)\, x_2(t)\, dt = 2\, \cos(s) + \frac{7}{2} - 2\, s^2 \\ \qquad + \cos(1)\, s^2 - 5\, \sin(1) - 3\, \cos(1) - 4\, \sin(1)\, s - 16\, s. \end{cases} \tag{2.29}$$

在Maple中输入下述命令:

```
exprs[5]:=x[1](s)-int((s+t)*x[1](t),t=0..1)-int((s)
        *x[2](t),t=0..1)=sin(s)+1/2-2*s+cos(1)*s-sin(1)
        +cos(1)-2*s*(sin(1)+4);
exprs[6]:=x[2](s)-int((s^2+t)*x[1](t),t=0..1)-int((2*s
        +2*t)*x[2](t),t=0..1)=2*cos(s)+7/2-2*s^2+cos(1)*s^2
        -5*sin(1)-3*cos(1)-4*sin(1)*s-16*s;
FredEqns([exprs[5],exprs[6]]);
```

即可得到下述机械化求解过程:

The system can be rewritten as:

$$X(s)-\lambda\int_0^2 K(s,t)X(s)dt = Y(s)$$

where

$$X(s) = (x_1(s), x_2(s-1)),$$

$$\begin{aligned} Y(s) = &(\sin(s) + \tfrac{1}{2} - 2s + \cos(1)s - \sin(1) + \cos(1) - 2s(\sin(1)+4),\\ &2\cos(s-1) + \tfrac{39}{2} - 2(s-1)^2 + \cos(1)(s-1)^2 - 5\sin(1) - 3\cos(1)\\ &-4\sin(1)(s-1) - 16s), \end{aligned}$$

$$K(s,t) = \begin{bmatrix} s+t & s \\ (s-1)^2+t & 2s-4+2t \end{bmatrix}$$

The Fredholm determinant of the kernel K(s,t) is as:

$$d(\lambda) = -\frac{37}{12}$$

and the Fredholm minor is:

$$D_\lambda(s,t) = \frac{8}{3}s - \frac{43}{72} + \frac{10}{9}t - \frac{7}{6}s^2 - \frac{8}{3}st + \frac{7}{6}ts^2$$

So, we can get the solution of the equation as:

$$(x_1(s), x_2(s)) = (1+\sin(s), 8+2\cos(s))$$

例 2.16: 求解:

$$\begin{cases} x_1(s) - \int_0^1 (s+t)\,x_1(t)\,dt - \int_0^1 s x_2(t)\,dt = -\frac{1}{2} + e^s - 3\,es + s, \\ x_2(s) - \int_0^1 (s+t)\,x_1(t)\,dt - \int_0^1 (2s+2t)\,x_2(t)\,dt = 2\,e^s - \frac{11}{2} - 5\,es + 2\,s. \end{cases} \tag{2.30}$$

在Maple中输入下述命令:

```
exprs[7]:=x[1](s)-int((s+t)*x[1](t),t=0..1)-int((s)
        *x[2](t),t=0..1)=-1/2+exp(s)-3*exp(1)*s+s;;
exprs[8]:=x[2](s)-int((s+t)*x[1](t),t=0..1)-1*int((2*s
        +2*t)*x[2](t), t=0..1)=2*exp(s)-11/2-5*exp(1)*s+2*s;
FredEqns([exprs[7],exprs[8]]);
```

则通过机械化算法可得到:

```
The system can be rewritten as:
```

$$X(s) - \lambda \int_0^2 K(s,t)X(s)dt = Y(s)$$

```
where
```

$$X(s) = (x_1(s), x_2(s-1)),$$

$$Y(s) = (\tfrac{1}{2} + e^s - 3\,es + s, 2\,e^{s-1} - \tfrac{15}{2} - 5\,e\,(s-1) + 2\,s),$$

$$K(s,t) = \begin{bmatrix} s+t & s \\ s-1+t & 2s-4+2t \end{bmatrix}$$

```
The Fredholm determinant of the kernel K(s,t) is as:
```

$$d(\lambda) = -\frac{11}{18}$$

```
and the Fredholm minor is:
```

$$D_\lambda(s,t) = -\frac{5}{6}s + \frac{73}{36} - \frac{17}{12}t + \frac{5}{6}st$$

```
So, we can get the solution of the equation as:
```

$$(x_1(s), x_2(s)) = (1 + e^s, 1 + 2e^s)$$

例 2.17: 求解:

$$\begin{cases} x_1(s) - \int_0^1 (s+t^2)\, x_2(t)\, dt = s^4 - \frac{7}{12} - \frac{3}{2}\, s, \\ x_2(s) - \int_0^1 (s^3+t)\, x_1(t)\, dt = s + \frac{5}{6} - \frac{1}{5}\, s^3. \end{cases} \tag{2.31}$$

在Maple中输入下述命令:

```
exprs[9]:=x[1](s)-int((s+t^2)*x[2](t),t=0..1)=s^4-7/12
        -3/2*s;
exprs[10]:=x[2](s)-int((s^3+t)*x[1](t),t=0..1)=s+5/6
        -1/5*s^3;
FredEqns([exprs[9],exprs[10]]);
```

则机械化算法FredEqns可输出下述结果:

```
The system can be rewritten as:
```

$$X(s) - \lambda \int_0^2 K(s,t)X(s)dt = Y(s)$$

```
where
```

$$X(s) = (x_1(s), x_2(s-1)),$$

$$Y(s) = (s^4 - \frac{7}{12} - \frac{3}{2}s, s - \frac{1}{6} - \frac{1}{5}(s-1)^3),$$

$$K(s,t) = \begin{bmatrix} 0 & s + (t-1)^2 \\ (s-1)^3 + t & 0 \end{bmatrix}$$

```
The Fredholm determinant of the kernel K(s,t) is as:
```

$$d(\lambda) = \frac{31}{144}$$

```
and the Fredholm minor is:
```

$$D_\lambda(s,t) = \frac{19}{48} + \frac{1}{2}(s-1)^3 + \frac{1}{2}(t-1)^2 + (s-1)^3(t-1)^2 + \frac{1}{6}s^2 - \frac{1}{6}s + \frac{1}{6}ts^3 - \frac{1}{18}s^3 - \frac{1}{12}s^3t^2 + \frac{1}{4}s^2t^2 + \frac{1}{2}st - \frac{1}{2}ts^2 - \frac{5}{24}t + \frac{5}{48}t^2 - \frac{1}{4}st^2$$

```
So, we can get the solution of the equation as:
```

$$(x_1(s), x_2(s)) = (s^4, s+1)$$

从上述五个算例可以看出, 机械化算法FredEqns的功能较为强大, 可以直接求出形如 (2.21) 的 Fredholm 积分方程组的解析解, 解中不仅可以包含多项式函数, 还可以包括更为复杂的正弦函数 sin, 余弦函数 cos 以及指数函数 exp 等. 但是, 由于 Fredholm 积分方程本身的复杂性, 随着 $k_{ij}(s,t)$ 的非线性的增强, 求解过程极为复杂, 在此不再例举.

2.3　本章小结

本章研究了 Fredholm 积分方程及其方程组的豫解核求解法, 在此基础上, 建立了基于Maple平台的机械化算法voltfredproc和FredEqns, 利用这些算法可以获得此类积分方程解析解. 应用算例表明, 上述算法是求解 Fredholm 积分方程的高效算法之一. 这将为 Fredholm 积分方程求解提供参考.

在基于豫解核方法求解 Fredholm 积分方程 (组) 的算法设计过程中, 困难的是计算 $d(\lambda)$ 和 $D_\lambda(s,t)$, 特别是 Fredholm 行列式的计算. 我们知道, 计算机代数系统Maple中有一个线性代数包linalg包含 100 多个函数, 可以解决线性代数的各种计算, 但无法直接计算 $d(\lambda)$ 和 $D_\lambda(s,t)$. 我们利用定理 2.1 以及矩阵的行列式的计算方法, 建立了计算 $d(\lambda)$ 和 $D_\lambda(s,t)$ 的子算法. 作为例子, 下面给出计算 Fredholm 行列式 $d(\lambda)$ 的算法Kdet的源码:

```
Kdet:=proc(n,k)
local v,i,j,d_det,D_det;
v:=Matrix(1..n):
for i from 1 to n do
    for j from 1 to n do
       v[i,j]:=k(s[i],s[j]);
    end do:
end do;
d_det:=simplify(det(v));
if n=1 then
    D_det:=0;
else
    D_det:=simplify(det(minor(v,2,1)));
end if;
return d_det,D_det;
end proc:
```

在上面的算法中, 命令det(A) 是用于计算矩阵 A 的行列式, 而minor(A,i,j) 返回元素 A_{ij} 的余子式, 即从 $\det(A)$ 中去除第 i 行第 j 列后剩余元素组成的行列式.

另外, 在Maple中有两个命令matrix 和Matrix 都用于建立矩阵, 但两者的数据结构完全不同. 例如, 我们要建立一个矩阵 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, 可以使用下述两种命令:

```
F:=matrix(2,2,[[a,b],[c,d]]);
G:=Matrix(2,2,[[a,b],[c,d]]);
```

但矩阵 F 的数据结构是symbol 而 G 的则是Matrix. 在实际应用中, 命令Matrix 较matrix 更实用, 因为我们可以通过相关命令获得Matrix 命令所建立的矩阵的基本信息, 诸如行数, 列数, 行元素, 列元素, 等等. 而由matrix 建立的矩阵由于其数据

结构为`symbol`, 我们无法获得进一步的信息, 因而无法进行更进一步的运算.

当然, 因为积分方程本身的复杂性以及`Maple`本身的一些特点, 我们从计算结果可以看出, `Maple`下的输出结果有时不符合我们的习惯 (例如排序的问题), 但这并不影响结果的正确性.

另外, 在机械化算法`fredholmproc`和`FredEqns`的设计过程中, 我们借助`lprint`和`print`给出了 Fredholm 积分方程 (组) 可读性的求解过程, 增加了求解过程的逻辑性和理解度, 这对其他类型方程的机械化求解的算法设计具有参考价值.

Fredholm 积分方程 (组) 机械化求解研究结果表明, 数学机械化是积分方程求解的重要方法之一. 其中最吸引人的当属利用计算机可以高效快捷地进行复杂计算. 但是, 我们绝不能因此而认为计算机是万能的, 它不能解决所有的方程求解问题. 对于更为复杂的非线性 Fredholm 积分方程的机械化求解依然是一个值得研究的领域.

第 3 章　Volterra 积分方程求解及其机械化

Volterra 积分方程是另一类经典的且应用广泛的积分方程. 自数学大师 Volterra 对于该类方程的奠基性工作开始, 关于 Volterra 积分方程求解就一直是积分方程研究的热点, 人们提出了各种各样的求解法 (详见 1.2 节). 但由于 Volterra 积分方程本身的复杂性, 所有的求解方法都存在着过程繁复、计算量大的问题. 为此, 本章将研究 Volterra 积分方程的求解及其机械化算法问题.

本章分为 4 节, 第 1 节, 研究 Neumann 级数法及其机械化求解 Volterra 积分方程的若干问题, 特别是探讨利用有限项迭代核通过数学归纳法得到通项公式并最终得到解析解的问题, 第 2 节研究 Taylor 级数求解法, 第 3 节, 研究 Volterra 积分方程组的求解, 最后, 也就是第 4 节, 给出 Volterra 积分方程机械化求解的一些讨论.

3.1　Neumann 级数求解法

第一章 (1.2 节) 简单综述了求解 Volterra 积分方程的各种方法, 在此不再赘述. 除了前已述及的方法之外, 还有经典的 Neumann 级数法 [212].

3.1.1　基本算法

首先, 我们重述求解第二种 Volterra 积分方程的 Neumann 级数法, 该方法是建立机械化算法的基础, 我们将详述此法.

考虑第二种 Volterra 积分方程形如:

$$x(s) - \lambda \int_a^s k(s,t)x(t)dt = y(s), \quad a \leqslant s \leqslant b. \tag{3.1}$$

其中, λ 是参数, 函数 $y(s) \in L^2[a,b]$ 及核 $k(s,t) \in L^2[a,b] \times [a,b]$ 已知, $x(s)$ 是要求解的函数.

为了便于叙述求解 (3.1) 的 Neumann 级数法, 我们先给出下面一些预备知识.

定义 3.1: 我们称 L^2- 函数序列 $\{x_n(s)\}(n=1,2,\cdots)$ 在 E 上几乎一致收敛于函数 $x(s)$, 如果存在一个非负 L^2- 函数 $p(s)$, 使得对于任给的 $\varepsilon > 0$, 存在与之对应的自然数 N, 当 $n \geqslant N$ 时, 有:

$$|x_n(s) - x(s)| \leqslant \varepsilon p(s).$$

引理 3.1: 假设 $k(s,t)\in L^2$, $a\leqslant t\leqslant s\leqslant b$, $x(s)\in L^2$, 令:

$$x_n(s)=\int_a^s k^n(s,t)x(t)dt \quad (n=1,2,\cdots)$$

则:

$$|x_n(s)|\leqslant \frac{k_1(s)\parallel x\parallel}{\sqrt{(n-1)!}}\left(\int_a^s [k_1(t)]^2\,dt\right)^{\frac{1}{2}(n-1)} \quad (n=1,2,\cdots) \tag{3.2}$$

这里,

$$\parallel x\parallel=\left(\int_t^b |k(s,t)|^2dt\right)^{\frac{1}{2}}$$

称为连续函数 $x(s)$ 的范数, 而

$$k_1(s)=\left(\int_a^s |k(s,t)|^2dt\right)^{\frac{1}{2}}.$$

引理 3.2: 假设 $k(s,t)\in L^2$, $a\leqslant t,s\leqslant b$, 则

$$|k^{n+1}(s,t)|\leqslant \frac{\parallel k\parallel^{n-1}}{\sqrt{(n-1)!}}k_1(s)k_2(s) \quad (n=1,2,\cdots) \tag{3.3}$$

其中,

$$\parallel k\parallel=\left(\int_a^s\int_a^s |k(s,t)|^2dsdt\right)^{\frac{1}{2}},$$

$$k_1(s)=\left(\int_a^s |k(s,t)|^2dt\right)^{\frac{1}{2}},$$

$$k_2(s)=\left(\int_t^b |k(s,t)|^2dt\right)^{\frac{1}{2}}.$$

基于上述结果, 我们可以得到求解第二种 Volterra 积分方程的 Neumann 级数法:

定理 3.1: 假设 $k(s,t)\in L^2$, 对于任意的 $y(s)\in L^2$ 和 $\lambda\in\mathbf{C}$, 方程 (3.1) 存在唯一的 L^2- 解, 即 Neumann 级数形式解:

$$x(s)=y(s)+\sum_{n=1}^{\infty}\lambda^n y_n(s), \tag{3.4}$$

其中

$$y_n(s)=\int_a^s k^n(s,t)y(t)dt \quad (n=1,2,\cdots), \tag{3.5}$$

这里, $k^n(s,t)$ 被称作 n 次迭代核 (iterative kernal), 满足:

$$k^n(s,t)=\int_t^s k^i(s,u)k^{n-i}(u,t)du \quad (n=2,3,\cdots;\ i=1,2,\cdots). \tag{3.6}$$

为了便于计算, 公式 (3.6) 可改写为下述形式:

$$k^n(s,t)=\int_t^s k(s,u)k^{n-1}(u,t)du \quad (n=2,3,\cdots,n-1). \tag{3.7}$$

这种改写在实际计算时非常有用.

另一方面, 根据定义 3.1, 容易验证, 方程 (3.1) 的 Neumann 级数解, 即 (3.4) 的右边表达式

$$y(s)+\sum_{n=1}^{\infty}\lambda^n y_n(s)$$

几乎一致收敛于 $x(s)$.

为了便于理解 Neumann 级数法求解 (3.1), 我们通过下面的例子列示求解过程.

例 3.1: 考虑方程:

$$x(s)-\int_0^s (s-t)x(t)dt=s,\ (0\leqslant s\leqslant b). \tag{3.8}$$

这里,

$$y(s)=s,\ k(s,t)=s-t,\ \lambda=1,$$

则:

$$\begin{aligned}
k^2(s,t)&=\textstyle\int_t^s k(s,u)k(u,t)du=\int_t^s(s-u)\cdot(u-t)dt=\frac{1}{6}(s-t)^3,\\
k^3(s,t)&=\textstyle\int_t^s k(s,u)k^2(u,t)du=\int_t^s(s-u)\cdot\frac{1}{6}(u-t)^3dt=\frac{1}{120}(s-t)^5,\\
k^4(s,t)&=\textstyle\int_t^s k(s,u)k^3(u,t)du=\int_t^s(s-u)\cdot\frac{1}{120}(u-t)^5dt=\frac{1}{5040}(s-t)^7,\\
k^5(s,t)&=\textstyle\int_t^s k(s,u)k^4(u,t)du=\int_t^s(s-u)\cdot\frac{1}{5040}(u-t)^7dt=\frac{1}{5040}(s-t)^9.
\end{aligned} \tag{3.9}$$

利用数学归纳法, 可得:

$$k^n(s,t)=\frac{1}{(2n-1)!}(s-t)^{(2n-1)}. \tag{3.10}$$

从而, 方程 (3.8) 的解为:

$$x(s) = y(s) + \int_0^s \sum_{n=1}^{\infty} \lambda^n k^n(s,t)dt = s + \int_0^s \sum_{n=1}^{\infty} \frac{1}{(2n-1)!}(s-t)^{(2n-1)}dt$$
$$= s + \int_0^s \sinh(s-t)dt = \sinh(s).$$

显然, 这是方程 (3.8) 的解析解.

在这个例子中, 我们看到由 (3.9) 得到 n 次迭代核 $k^n(s,t)$ (3.10) 用到了数学归纳法. 这是利用 Neumann 级数法求解方程 (3.1) 能够得到解析解的最为关键的一步, 同时也是实现机械化求解 Volterra 积分方程至关重要的一步. 关于数学归纳法自动推证的细节将在 3.1.4 节给出.

另外, 通过这个例子, 我们也看到了 Neumann 级数法求解方程 (3.1) 同样存在计算量大且求解过程繁复的问题. 这也正是我们在Maple下建立机械化算法的初衷.

3.1.2 Neumann 级数法的机械化算法

本节的主要目的是在Maple下建立基于 Neumann 级数法求解方程 (3.1) 的机械化算法, 利用该算法可以方便、快捷地获取方程 (3.1) 的解析解.

从上节研究可以看出, 用于求解方程 (3.1) 的公式 (3.4)— (3.7) 提供了在Maple下建立机械化算法的基本方法, 我们根据这些公式可以在Maple下进行相应的算法设计和程序设计. 机械化算法voltproc 的设计思想参见下图 (图 3.1).

而当我们利用机械化算法voltproc 求解方程 (3.1) 时, 所要做的只是输入关于方程的信息, 机械化算法则会提供方程 (3.1) 的解析解以及求解过程 — 如第二章所述, 我们仍然选择了求解过程的可读性输出, 以便于人们理解机械化算法的求解过程. 机械化算法的主程序voltproc的源码列示如下:

```
voltproc:=proc(expression)
local k, fa, f, N, i, k_n, Flag, temp, H, par, lambda1,
          xfunction, yfunction, key, expr, bounds, fy;
N:=10;
par:=parmeint(expression); lambda1:=par[1];
expr:=par[2]; xfunction:=par[3]; yfunction:=par[4];
bounds:=par[5]; fy:=unapply(yfunction, s);
k:=klist(expr,N);
fa:=factorlist(k);
temp:=induction(k,fa);
f:=temp[1];
Flag:=temp[2];
```

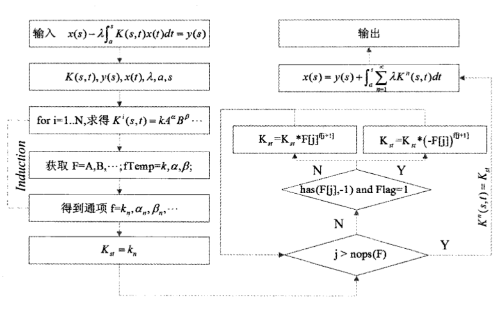


图 3.1 Neumann 级数法求解 Volterra 积分方程的机械化算法流程图

```
k_n:=1/rhs(f[1])!;
for i from 1 to nops(fa) do
    if type(fa[i],polynom) and Flag=1 and has(fa[i],-1)
      then k_n:=k_n*(-fa[i])^rhs(f[i+1]);
      Flag:=0;
    else
      k_n:=k_n*(fa[i])^rhs(f[i+1]);
    fi;
od;
H:=simplify(sum(lambda1^n*k_n,n=0..infinity));
key:=simplify(yfunction+lambda1*int(H*fy(t),t=bounds[1]
          ..bounds[2]));
lprint('Here, the parameters of the system are as
          follows:');
print('y(s)'=yfunction,'lambda'=lambda1,'k(s,t)'=expr);
lprint('The nth term of the iterative kernel k^n(s,t) is
          as follows:');
```

```
print('k^n'(s,t)=k_n);
lprint(`The exact solution of the system is as follows:`);
print('x(s)'=key);
end:
```

这里, 主程序voltproc中的参数expression表示要求解的方程的信息. 例如, 对于方程 (3.1), 我们在Maple下需要输入的仅仅是以下信息:

```
expr:=x(s)=lambda*int(k(s,t)*x(t),t=a..s)+y(s);
voltproc(expr);
```

则机械化算法voltproc即可输出该方程的解析解.

另一方面, 子程序parmeint 的主要功能就是提取方程 (2.10) 中的上述信息, 包括参数 λ, 核函数 $k(s,t)$, 函数 $y(s)$, 积分限 $[0,s]$ 以及待求解函数 $x(s)$. 而induction 正是我们设计的数学归纳法的机械化算法, 由于其在自动化推理中的重要性, 将在 3.1.4 详述.

另外, 在voltproc中我们取定 $N=10$, 即我们利用公式 (3.7) 分别计算 $k^2(s,t)$, $k^3(s,t)$, $\cdots$, $k^{10}(s,t)$, 然后再利用数学归纳法获取 n 次迭代核的通项公式 $k^n(s,t)$. 这里, 我们选取 $N=10$ 并无什么特殊含义, 也可取其他的数值, 例如 $N=8$ 或 20. 但 N 太小我们可能很难得到通项公式, 而 N 太大则使计算复杂性大幅提高且没有必要.

3.1.3 应用算例

本节, 我们将通过几个例子展示基于 Neumann 级数法的机械化求解 Volterra 积分方程的过程.

例 3.2: 求解方程:

$$x(s)=e^s+\int_0^s e^{s-t}x(t)dt. \tag{3.11}$$

在Maple中, 我们需要输入关于方程 (3.11) 的信息:

```
expr2:=x(s)=exp(s)+int(exp(s-t)*x(t),t=0..s);
voltproc(expr2);
```

则机械化算法voltproc即可输出整个求解过程及其解析解:

```
Here,the parameters of the system are as follows:
```

$$y(s)=e^s,\ \lambda=1,\ k(s,t)=e^{s-t}$$

The nth term of the iterative kernel $k^n(s,t)$ is as follows:

$$K^n(s,t) = \frac{e^s(s-t)^n}{n!\,e^t}$$

The exact solution of the system is as follows:

$$x(s) = e^{(2s)}$$

容易验证 $x(s) = e^{2s}$ 即为方程 (3.11) 的解析解. 而整个求解过程只需 2 秒即可完成.

例 3.3: 解方程:

$$x(s) = s3^s - \int_0^s 3^{s-t}x(t)dt. \tag{3.12}$$

输入命令:

```
expr3:=x(s)=s*3^s-int(3^(s-t)*x(t),t=0..s);
voltproc(expr3);
```

即可得机械化求解过程:

Here,the parameters of the system are as follows:

$$y(s) = s3^s,\ \lambda = -1,\ k(s,t) = 3^{s-t}$$

The nth term of the iterative kernel $k^n(s,t)$ is as follows:

$$K^n(s,t) = \frac{3^s(s-t)^n}{n!\,3^t}$$

The exact solution of the system is as follows:

$$x(s) = -3^s(-1+e^{-s})$$

显然, 这是方程 (3.12) 的解析解.

例 3.4: 解

$$x(s) = 1 - 2\,s - \int_0^s e^{s^2-t^2}x(t)dt. \tag{3.13}$$

输入命令:

```
expr4:=x(s)=1-2*s-int(exp(s^2-t^2)*x(t),t=0..s);
```

```
voltproc(expr4);
```

则机械化求解过程为:

```
Here,the parameters of the system are as follows:
```

$$y(s) = 1 - 2\,s,\ \lambda = -1,\ k(s,t) = e^{s^2 - t^2}$$

```
The nth term of the iterative kernel kn(s,t) is as follows:
```

$$K^n(s,t) = \frac{e^{s^2}(s-t)^n}{n!\,e^{t^2}}$$

```
The exact solution of the system is as follows:
```

$$x(s) = -2\,s + e^{(s-1)s}$$

此即解析解.

例 3.5: 考虑方程:

$$x(s) = 1 + s^2 + \int_0^s \frac{1+s^2}{1+t^2} x(t)dt. \tag{3.14}$$

在Maple下输入:

```
expr5:=x(s)=1+s^2+int((1+s^2)/(1+t^2)*x(t),t=0..s);
voltproc(expr5);
```

即可得求解过程如下:

```
Here,the parameters of the system are as follows:
```

$$y(s) = 1 + s^2,\ \lambda = 1,\ k(s,t) = \frac{1+s^2}{1+t^2}$$

```
The nth term of the iterative kernel kn(s,t) is as follows:
```

$$K^n(s,t) = \frac{(1+s^2)(s-t)^n}{n!\,(1+t^2)}$$

```
The exact solution of the system is as follows:
```

$$x(s) = (1+s^2)e^s$$

为方程 (3.13) 的解析解.

例 3.6: 解方程:

$$x(s) - 2\int_1^s e^{s^2-t^2} x(t)dt = e^{s^2+2s}. \tag{3.15}$$

输入命令:

```
expr6:=x(s)-2*int(exp(s^2-t^2)*x(t),t=1..s)=exp(s^2+2*s);
voltproc(expr6);
```

即可得求解过程为:

Here,the parameters of the system are as follows:

$$y(s) = e^{s^2+2s},\ \lambda = 2,\ k(s,t) = e^{s^2-t^2}$$

The nth term of the iterative kernel $k^n(s,t)$ is as follows:

$$K^n(s,t) = \frac{e^{s^2}(s-t)^n}{n!\, e^{t^2}}$$

The exact solution of the system is as follows:

$$x(s) = e^{s(s+2)}(-1+2\,s)$$

即为 (3.15) 的解析解.

例 3.7: 解方程

$$x(s) = \sin s + 2\int_0^s e^{s-t} x(t)dt. \tag{3.16}$$

由命令:

```
expr7:=x(s)=sin(s)+2*int(exp(s-t)*x(t),t=0..s);
voltproc(expr7);
```

可得机械化求解过程为:

Here,the parameters of the system are as follows:

$$y(s) = \sin s,\ \lambda = 2,\ k(s,t) = e^{s-t}$$

The nth term of the iterative kernel $k^n(s,t)$ is as follows:

$$K^n(s,t) = \frac{e^s(s-t)^n}{n!\, e^t}$$

```
The exact solution of the system is as follows:
```

$$x(s)=\frac{2}{5}\sin s-\frac{1}{5}\cos s+\frac{1}{5}e^{3s}$$

此为 (3.16) 的解析解.

例 3.8: 求解:

$$x(s)=e^{s}\sin s+\int_{0}^{s}\frac{2+\cos s}{2+\cos t}x(t)dt. \tag{3.17}$$

输入命令:

```
expr8:=x(s)=exp(s)*sin(s)+int((2+cos(s))/(2+cos(t))*x(t),
        t=0..s);
voltproc(expr8);
```

即得求解过程:

```
Here,the parameters of the system are as follows:
```

$$y(s)=e^{s}\sin s,\ \lambda=1,\ k(s,t)=\frac{2+\cos s}{2+\cos t}$$

The nth term of the iterative kernel $k^{n}(s,t)$ is as follows:

$$K^{n}(s,t)=\frac{(s-t)^{n}(2+\cos(s))}{n!\,(2+\cos(t))}$$

```
The exact solution of the system is as follows:
```

$$x(s)=e^{s}\left(\sin s-2\,\ln(2+\cos s)-\ln(2+\cos s)\cos s+2\,\ln 3+\ln 3\cos s\right)$$

此为 (3.17) 的解析解.

下面, 我们给出一个积分方程中含有未定函数的求解问题.

例 3.9: 解方程

$$x(s)=\lambda\int_{a}^{s}e^{s-t}x(t)dt+f(s). \tag{3.18}$$

在Maple下输入:

```
expr9:=x(s)=lambda*int(exp(s-t)*x(t),t=a..s)+f(s);
voltproc(expr9);
```

则机械化算法voltproc输出求解过程:

Here,the parameters of the system are as follows:

$$y(s) = f(s),\ \lambda = \lambda,\ k(s,t) = e^{s-t}$$

The nth term of the iterative kernel $k^n(s,t)$ is as follows:

$$K^n(s,t) = \frac{e^s(s-t)^n}{n!\,e^t}$$

The exact solution of the system is as follows:

$$x(s) = f(s) + \lambda \int_0^s e^{(1+\lambda)(s-t)} f(t)dt$$

即为 (3.18) 的解. 显然, 当 $f(t)$ 给定时, 我们能够获取方程 (3.18) 的解析解.

3.1.4 数学归纳法机械化推理的进一步探讨

上节, 基于 Neumann 级数法并结合数学归纳法建立了求解第二种 Volterra 积分方程的机械化算法voltproc, 该算法最大的特点是能够提供积分方程的解析解, 而其中最重要的一个环节是通过前 N 个迭代核 $k^N(s,t)$ 的计算结果 (在算法voltproc中, 我们取 $N = 10$), 运用数学归纳法得到 n 次迭代核 $k^n(s,t)$ 的通项公式, 然后再利用公式 (3.4) 即可得到所求方程的解析解.

众所周知, 数学归纳法是一种在定理自动推证和机械化求解中具有重要作用的数学方法, 其最大的难点在于利用有限项信息获得通项公式. 一般情况下, 我们获取通项的方法是基于经验或者简单的推导, 而一般的通用算法似乎是不存在的, 因而也不存在可用于计算机自动推证的通用算法. 在算法induction设计过程中, 我们成功地利用 "分解" 原理将 $k^N(s,t)$ 分解为若干个部分, 这一分解过程由子程序klist完成:

```
klist:=proc(expr,n)
local temp,temp2,temp1,k,i;
temp:=unapply(expr,s,t); temp2:=temp(u,t);
k[1]:=factor(temp(s,t));
for i from 2 to n do
    temp1:=int(temp(s,u)*temp2,u=t..s);
    temp1:=unapply(temp1,s,t);
    k[i]:=factor(temp1(s,t));
    if type(k[i],'+') then return FALL fi;
    temp2:=temp1(u,t);
```

```
od;
k:=convert(k,list);
k:=subsop(1=NULL,k);
return k;
end:
```

在此基础上, 对每一部分逐一运用数学归纳法, 最后再按照原来 “分解” 的逻辑顺序合并在一起, 从而得到了计算通项 $k^n(s,t)$ 的机械化算法. 下面, 我们给出数学归纳法算法induction的Maple源代码:

```
with(stats):
induction:=proc(List::list)
local num,i,Later,k,temp,Flag1,Flag2,f,Flag,y;
num:=nops(List):  Later:=[ ];
Flag1:=0;
Flag2:=0;
Flag:=0;
for i from 1 to num do
    k:=1;
    if i mod 2=0 and List[i]<0 then Flag1:=1;fi;
    if i mod 2<>0 and List[i]<0 then Flag2:=1;fi;
    if abs(List[i])<1 then
       Flag:=1;
       temp:=abs(1/List[i]);
    elif abs(List[i])>=1 then
       temp:=abs(List[i]);
    fi;
    while temp<>1 do
       temp:=temp/k;
       k:=k+1;
       if temp<1 then return ERROR fi;
    od;
    if i=1 and k=1 then
       Later:=[op(Later),k];
    else
```

```
        Later:=[op(Later),k-1];
    fi;
od;
f:=fit[leastsquare[[n,y]]]([[seq(i,i=1..num)],
        [seq(Later[j],j=1..num)]]);
if Flag1=1 and Flag2=1 then
    y:=-rhs(f)!;
elif Flag1=1 and Flag2=0 then
    y:=(-1)^(n)*rhs(f)!;
elif Flag1=0 and Flag1=1 then
    y:=(-1)^(n)*rhs(f)!;
else
    y:=rhs(f);
fi;
if Flag=1 then
    y:=1/y;
else
    y:=y!;
fi;
lprint('The nth term of the sequence is as follows:')
print(y);
end:
```

尽管如此, 但我们想强调的是可能由于 Neumann 级数解的固有优点使得我们的数学归纳法得以成功应用, 而当我们将这一做法用于其他问题的解决时, 却因为重重困难而无法得到我们期望的结果. 这也将是我们后续研究的一个问题.

3.2 Taylor 级数求解法

Taylor 级数法是一种重要的方程求解方法, 在微分方程 (包括常微分方程, 偏微分方程等) 领域有广泛应用. 但 Taylor 级数法运算结果的精确度由级数展开的阶数决定. 要想获取较高的精度必须选择较大的阶数, 而这势必又带来计算量大、运算过程繁复的问题. 因此, 借助计算机大容量、高精度、高效率的特点在计算机代数系统下建立相应的机械化算法是必要的.

本节的内容安排为: 首先, 我们给出求解 Volterra 积分方程的 Taylor 级数法原理, 在此基础上建立相应的机械化算法, 最后, 给出若干算例说明机械化求解算法的应用.

3.2.1　求解 Volterra 积分方程的 Taylor 级数法

考虑下述第二种 Volterra 积分方程:

$$x(s)-\lambda\int_0^s k(s,t)x(t)dt=y(s),\quad 0\leqslant s\leqslant 1. \tag{3.19}$$

这里, 参数 λ 和函数 $k(s,t)$, $y(s)$ 均已知, 而 $x(s)$ 是待求解的函数.

将方程 (3.19) 中的 $x(s)$ 在 $s=0$ 处展开成 Taylor 级数:

$$x(s)=\sum_{i=0}^{\infty}a_is^i=a_0+a_1s+a_2s^2+\cdots+a_ns^n+R_n(s). \tag{3.20}$$

其中 $R_n(s)$ 为余项, a_i $(i=0,1,\cdots,n)$ 是待定系数.

略去余项 $R_n(s)$, 即将 (3.20) 转换为 Taylor 多项式形式:

$$x(s)=a_0+a_1s+a_2s^2+\cdots+a_ns^n. \tag{3.21}$$

接下来, 用同样的方法得到 $y(s)$ (在 $s=0$ 点) 和 $k(s,t)$ (在 $t=0$ 点) 的 Taylor 多项式结果. 将这些结果全部代入方程 (3.19), 则方程 (3.19) 中的积分部分可积. 我们就可把积分方程 (3.19) 转换成了关于参数 s 的含有 $n+1$ 个未知参数 a_i $(i=0,1,\cdots,n)$ 的形式, 然后提取 s^i $(i=0,1,\cdots,n)$ 的系数即可得到 a_i $(i=0,1,\cdots,n)$ 的代数方程组. 求解该代数方程组可得到系数 a_i $(i=0,1,\cdots,n)$, 即可获得积分方程的 Taylor 多项式解 (3.21).

显然, Taylor 多项式解 (3.21) 的精度依赖于 (3.20) 中级数的展开阶数 n, n 越大精度越高, 相应的计算复杂度也迅速提高. 下节, 将借助计算机代数系统Maple建立相应的机械化算法.

3.2.2　Taylor 级数法的机械化算法

Taylor 级数法求解 Volterra 积分方程的近似精确解的过程可以在计算机代数系统Maple下建立相应的机械化算法. 机械化算法的主程序mainproc的Maple源代码如下所示:

```
mainproc:=proc(expr, N)
local Expr,parme,lambda,k,xfunction,yfunction,bounds,
      Xfunction,Func,TaylorT,TaylorS,eqn,key,i,result;
```

```
result:=0;
Expr:=Union(expr);
parme:=parmeint(Expr);
lambda:=parme[1]; k:=parme[2];
xfunction:=parme[3];
yfunction:=parme[4];
bounds:=parme[5];
xfunction:=sum(a[i]*s^i,i=0..N+1);
Xfunction:=unapply(xfunction,s);
TaylorT:=convert(taylor(k,s=0,N+1),polynom);
TaylorS:=convert(taylor(TaylorT,t=0,N+1),polynom);
Func:=sort(collect(Xfunction(s)-convert(taylor(yfunction,
        s=0,N+1),polynom)-lambda*int(collect(expand(TaylorS
        *Xfunction(t)),t),t=0..s),s),s);
lprint('The equation can be converted into the following
        algebraic equations:');
eqn:=seq(coeff(Func,s,i)=0,i=0..N):
print(eqn);
key:=solve(eqn,seq(a[i],i=0..N));
lprint('Solve this algebraic equations, we can get:');
print(key);
for i from 1 to N+1 do
    result:=result+rhs(key[i])*s^op(lhs(key[i]));
od;
lprint('Then the solution of the equation is as follows:');
print(parme[3]=result);
end proc:
```

为了便于计算, 在主程序mainproc中我们取 s 作为约束变量, 即要求输入方程时取 s 为变量, 其他参数集为:

expr: 要求解的积分方程;
N: Taylor 多项式解的阶数.

例如, 对于方程

$$x(s) = e^s \sin s + \int_0^s e^{s-t} x(t)dt.$$

在Maple中, 我们需要输入以下信息:

```
expr:=x(s)=exp(s)*sin(s)+int(exp(s-t)*x(t),t=0..s);

mainproc(expr, 10);
```

即可获取该积分方程的 10 阶 Taylor 近似解.

另一方面, 主程序mainproc中的两个子程序Union和parmeint的作用或源码参见 2.1.2 所述.

3.2.3　应用算例

为了更清楚地理解mainproc机械化求解过程, 本节, 我们选择了 8 个不同类型的例子, 其中, 5 个第二种 Volterra 积分方程 —3 个线性和 2 个非线性, 2 个第一种 Volterra 积分方程 —1 个线性 1 个非线性, 还有 1 个例子是积分 — 微分方程. 首先给出的是一个简单的线性 Volterra 积分方程例子.

例 3.10: 考虑第二种 Volterra 积分方程

$$x(s) = \cos s - e^s \sin s + \int_0^s e^s x(t)dt. \tag{3.22}$$

在Maple中, 我们需要输入关于方程的下述信息:

```
expr1:=x(s)=cos(s)-exp(s)*sin(s)+int(exp(s)*x(t),t=0..s);

mainproc(expr1,10);
```

则机械化算法mainproc立即返回以下结果:

The equation can be converted into the following algebraic equations:

$$a_0 - 1 = 0$$

$$a_1 - a_0 + 1 = 0$$

$$a_2 - \frac{1}{2}a_1 - a_0 + \frac{3}{2} = 0$$

$$-\frac{1}{3}a_2 - \frac{1}{2}a_1 + a_3 - \frac{1}{2}a_0 + \frac{1}{3} = 0$$

$$a_4 - \frac{1}{4}a_3 - \frac{1}{3}a_2 - \frac{1}{4}a_1 - \frac{1}{6}a_0 - \frac{1}{24} = 0$$

$$a_5 - \frac{3}{5}a_4 - \frac{1}{4}a_3 - \frac{1}{6}a_2 - \frac{1}{12}a_1 - \frac{1}{24}a_0 - \frac{1}{30} = 0$$

$$a_6 - \frac{1}{6}a_5 - \frac{1}{5}a_4 - \frac{1}{8}a_3 - \frac{1}{18}a_2 - \frac{1}{48}a_1 - \frac{1}{120}a_0 - \frac{7}{720} = 0$$

$$a_7 - \frac{1}{6}a_6 - \frac{1}{6}a_5 - \frac{1}{10}a_4 - \frac{1}{24}a_3 - \frac{1}{72}a_2 - \frac{1}{240}a_1 - \frac{1}{720}a_0 - \frac{1}{630} = 0$$

$$a_8 - \frac{1}{8}a_7 - \frac{1}{7}a_6 - \frac{1}{12}a_5 - \frac{1}{30}a_4 - \frac{1}{96}a_3 - \frac{1}{360}a_2 - \frac{1}{1440}a_1 - \frac{1}{5040}a_0 - \frac{1}{40320} = 0$$

$$a_9 - \frac{1}{9}a_8 - \frac{1}{8}a_7 - \frac{1}{14}a_6 - \frac{1}{36}a_5 - \frac{1}{120}a_4 - \frac{1}{480}a_3 - \frac{1}{2160}a_2 - \frac{1}{10080}a_1 - \frac{1}{40320}a_0 + \frac{1}{22680} = 0$$

$$a_{10} - \frac{1}{10}a_9 - \frac{1}{9}a_8 - \frac{1}{16}a_7 - \frac{1}{42}a_6 - \frac{1}{144}a_5 - \frac{1}{600}a_4 - \frac{1}{2880}a_3 - \frac{1}{15120}a_2 - \frac{1}{80640}a_1 - \frac{1}{362880}a_0 + \frac{11}{1209600} = 0$$

Solve this algebraic equations, we can get:

$$\{a_{10} = -\frac{1}{3628800}, a_9 = 0, a_8 = \frac{1}{40320}, a_7 = 0, a_6 = -\frac{1}{720}, a_5 = 0, a_4 = \frac{1}{24}, a_3 = 0, a_2 = -\frac{1}{2}, a_1 = 0, a_0 = 1\}$$

Then the solution of the equation is as follows:

$$x(s) = 1 - \frac{1}{2}s^2 + \frac{1}{24}s^4 - \frac{1}{720}s^6 + \frac{1}{40320}s^8 - \frac{1}{3628800}s^{10}$$

事实上, 方程 (3.22) 的解析解为:

$$x(s) = \cos s = 1 - \frac{1}{2}s^2 + \frac{1}{24}s^4 - \frac{1}{720}s^6 + \frac{1}{40320}s^8 - \frac{1}{3628800}s^{10} + \cdots$$

显然, 机械化算法`mainproc`返回的是方程 (3.22) 的解析解 $\cos s$ 的 10 阶 Taylor 多项式解. 这里的 10 是我们在运行主程序`mainproc`时输入的第 2 个参数值. 如果我们想得到精度更高的结果, 我们需要选取更大的 N, 即增大`mainproc` 的第 2 个参数值.

例 3.11: 求解第二种 Volterra 积分方程:

$$x(s) = \sin s + 2\int_0^s e^{s-t}x(t)dt. \tag{3.23}$$

输入命令:

```
expr2:=x(s)=sin(s)+2*int(exp(s-t)*x(t),t=0..s);
mainproc(expr2,10);
```

即可得 10 阶 Taylor 多项式解的求解过程为:

```
The equation can be converted into the following algebraic equations:
```

$$a_0 = 0$$

$$a_1 - 2\,a_0 - 1 = 0$$

$$a_2 - a_1 - a_0 = 0$$

$$a_3 - \tfrac{2}{3}\,a_2 - \tfrac{1}{3}\,a_1 - \tfrac{1}{3}\,a_0 + \tfrac{1}{6} = 0$$

$$a_4 - \tfrac{1}{2}\,a_3 - \tfrac{1}{6}\,a_2 - \tfrac{1}{12}\,a_1 - \tfrac{1}{12}\,a_0 = 0$$

$$a_5 - \tfrac{2}{5}\,a_4 - \tfrac{1}{10}\,a_3 - \tfrac{1}{30}\,a_2 - \tfrac{1}{60}\,a_1 - \tfrac{1}{60}\,a_0 - \tfrac{1}{120} = 0$$

$$a_6 - \tfrac{1}{3}\,a_5 - \tfrac{2}{15}\,a_4 - \tfrac{1}{60}\,a_3 - \tfrac{1}{180}\,a_2 - \tfrac{1}{360}\,a_1 - \tfrac{1}{360}\,a_0 = 0$$

$$a_7 - \tfrac{2}{7}\,a_6 - \tfrac{1}{21}\,a_5 - \tfrac{1}{105}\,a_4 - \tfrac{1}{420}\,a_3 - \tfrac{1}{1260}\,a_2 - \tfrac{1}{2520}\,a_1 - \tfrac{1}{2520}\,a_0 + \tfrac{1}{5040} = 0$$

$$a_8 - \tfrac{1}{4}\,a_7 - \tfrac{1}{28}\,a_6 - \tfrac{1}{168}\,a_5 - \tfrac{1}{840}\,a_4 - \tfrac{1}{3360}\,a_3 - \tfrac{1}{10080}\,a_2 - \tfrac{1}{20160}\,a_1 - \tfrac{1}{20160}\,a_0 = 0$$

$$a_9 - \tfrac{2}{9}\,a_8 - \tfrac{1}{36}\,a_7 - \tfrac{1}{252}\,a_6 - \tfrac{1}{1512}\,a_5 - \tfrac{1}{7560}\,a_4 - \tfrac{1}{30240}\,a_3 - \tfrac{1}{90720}\,a_2 - \tfrac{1}{181440}\,a_1 - \tfrac{1}{181440}\,a_0 - \tfrac{1}{362880} = 0$$

$$a_{10} - \tfrac{1}{5}\,a_9 - \tfrac{1}{45}\,a_8 - \tfrac{1}{360}\,a_7 - \tfrac{1}{2520}\,a_6 - \tfrac{1}{15120}\,a_5 - \tfrac{1}{75600}\,a_4 - \tfrac{1}{302400}\,a_3 - \tfrac{1}{907200}\,a_2 - \tfrac{1}{1814400}\,a_1 - \tfrac{1}{1814400}\,a_0 = 0$$

```
Solve this algebraic equations, we can get:
```

$$\{a_0 = 0, a_1 = 1, a_2 = 1, a_3 = \tfrac{5}{6}, a_4 = \tfrac{2}{3}, a_5 = \tfrac{49}{120}, a_6 = \tfrac{73}{360}, a_7 = \tfrac{437}{5040}, a_8 = \tfrac{41}{1260}, a_9 = \tfrac{3937}{362880}, a_{10} = \tfrac{1181}{362880}\}$$

```
Then the solution of the equation is as follows:
```

$$x(s) = s + s^2 + \frac{5}{6}s^3 + \frac{2}{3}s^4 + \frac{49}{120}s^5 + \frac{73}{360}s^6 + \frac{437}{5040}s^7 + \frac{41}{1260}s^8 + \frac{3937}{362880}s^9 + \frac{1181}{362880}s^{10}$$

易证, 方程 (3.23) 的解析解为:

$$x(s) = \tfrac{1}{5}\, e^{3s} - \tfrac{1}{5}\, \cos s + \tfrac{2}{5}\, \sin s$$
$$= s+s^2+\tfrac{5}{6}s^3+\tfrac{2}{3}s^4+\tfrac{49}{120}s^5+\tfrac{73}{360}s^6+\tfrac{437}{5040}s^7+\tfrac{41}{1260}s^8+\tfrac{3937}{362880}s^9+\tfrac{1181}{362880}s^{10}+\cdots$$

例 3.12: 解第二种 Volterra 积分方程:

$$x(s) = 1 + s^2 + \int_0^s \frac{1+s^2}{1+t^2}\, x(t)\, dt. \tag{3.24}$$

输入命令:

```
expr3:=x(s)=1+s^2+int((1+s^2)/(1+t^2)*x(t),t=0..s);
mainproc(expr3,10);
```

即可得求解过程为:

```
The equation can be converted into the following algebraic equations:
```

$$a_0 - 1 = 0$$
$$a_1 - a_0 = 0$$
$$a_2 - \tfrac{1}{2}\, a_1 - 1 = 0$$
$$a_3 - \tfrac{1}{3}\, a_2 - \tfrac{2}{3}\, a_0 = 0$$
$$a_4 - \tfrac{1}{4}\, a_3 - \tfrac{1}{4}\, a_1 = 0$$
$$a_5 - \tfrac{1}{5}\, a_4 - \tfrac{2}{15}\, a_2 + \tfrac{2}{15}\, a_0 = 0$$
$$a_6 - \tfrac{1}{6}\, a_5 - \tfrac{1}{12}\, a_3 + \tfrac{1}{12}\, a_1 = 0$$
$$a_7 - \tfrac{1}{7}\, a_6 - \tfrac{2}{35}\, a_4 + \tfrac{2}{35}\, a_2 - \tfrac{2}{35}\, a_0 = 0$$
$$a_8 - \tfrac{1}{8}\, a_7 - \tfrac{1}{24}\, a_5 + \tfrac{1}{24}\, a_3 - \tfrac{1}{24}\, a_1 = 0$$
$$a_9 - \tfrac{1}{9}\, a_8 - \tfrac{2}{63}\, a_6 + \tfrac{2}{63}\, a_4 - \tfrac{2}{63}\, a_2 + \tfrac{2}{63}\, a_0 = 0$$
$$a_{10} - \tfrac{1}{10}\, a_9 - \tfrac{1}{40}\, a_7 + \tfrac{1}{40}\, a_5 - \tfrac{1}{40}\, a_3 + \tfrac{1}{40}\, a_1 = 0$$

$$a_{11} + \tfrac{2}{99}\, a_2 - \tfrac{2}{99}\, a_8 - \tfrac{2}{99}\, a_4 - \tfrac{2}{99}\, a_0 + \tfrac{2}{99}\, a_6 - \tfrac{1}{11}\, a_{10} = 0$$
$$a_{12} + \tfrac{1}{60}\, a_3 - \tfrac{1}{60}\, a_5 - \tfrac{1}{60}\, a_9 + \tfrac{1}{60}\, a_7 - \tfrac{1}{60}\, a_1 - \tfrac{1}{12}\, a_{11} = 0$$
$$a_{13} + \tfrac{2}{143}\, a_8 - \tfrac{2}{143}\, a_2 - \tfrac{2}{143}\, a_{10} - \tfrac{1}{13}\, a_{12} - \tfrac{2}{143}\, a_6 + \tfrac{2}{143}\, a_4 + \tfrac{2}{143}\, a_0 = 0$$
$$a_{14} - \tfrac{1}{84}\, a_7 - \tfrac{1}{14}\, a_{13} + \tfrac{1}{84}\, a_1 + \tfrac{1}{84}\, a_5 - \tfrac{1}{84}\, a_3 + \tfrac{1}{84}\, a_9 - \tfrac{1}{84}\, a_{11} = 0$$
$$a_{15} + \tfrac{2}{195}\, a_6 + \tfrac{2}{195}\, a_{10} - \tfrac{2}{195}\, a_8 - \tfrac{2}{195}\, a_{12} - \tfrac{2}{195}\, a_4 + \tfrac{2}{195}\, a_2 - \tfrac{2}{195}\, a_0 - \tfrac{1}{15}\, a_{14} = 0$$

Solve this algebraic equations, we can get:

$$\{a_0=1,\ a_1=1,\ a_2=\tfrac{3}{2},\ a_3=\tfrac{7}{6},\ a_4=\tfrac{13}{24},\ a_5=\tfrac{7}{40},\ a_6=\tfrac{31}{720},\ a_7=\tfrac{43}{5040},$$
$$a_8=\tfrac{19}{13440},\ a_9=\tfrac{73}{362880},\ a_{10}=\tfrac{13}{518400},\ a_{11}=\tfrac{37}{13305600},\ a_{12}=\tfrac{19}{68428800},$$
$$a_{13}=\tfrac{157}{6227020800},\ a_{14}=\tfrac{61}{29059430400},\ a_{15}=\tfrac{211}{1307674368000}\}$$

Then the solution of the equation is as follows:

$$x(s)=1+s+\tfrac{3}{2}s^2+\tfrac{7}{6}s^3+\tfrac{13}{24}s^4+\tfrac{7}{40}s^5+\tfrac{31}{720}s^6+\tfrac{43}{5040}s^7+\tfrac{19}{13440}s^8+\tfrac{73}{362880}s^9+\tfrac{13}{518400}s^{10}$$
$$+\tfrac{37}{13305600}s^{11}+\tfrac{19}{68428800}s^{12}+\tfrac{157}{6227020800}s^{13}+\tfrac{61}{29059430400}s^{14}+\tfrac{211}{1307674368000}s^{15}$$

容易验证方程 (3.24) 的解析解为:

$$x(s)=e^s(1+s^2)$$
$$=1+s+\tfrac{3}{2}s^2+\tfrac{7}{6}s^3+\tfrac{13}{24}s^4+\tfrac{7}{40}s^5+\tfrac{31}{720}s^6+\tfrac{43}{5040}s^7+\tfrac{19}{13440}s^8+\tfrac{73}{362880}s^9+\tfrac{13}{518400}s^{10}$$
$$+\tfrac{37}{13305600}s^{11}+\tfrac{19}{68428800}s^{12}+\tfrac{157}{6227020800}s^{13}+\tfrac{61}{29059430400}s^{14}+\tfrac{211}{1307674368000}s^{15}$$
$$+\cdots$$

上述三个例子均属于第二种 Volterra 线性积分方程. 接下来的两个例子是非线性的情形. 我们也可以通过这两个例子进一步了解非线性积分方程的计算复杂性.

例 3.13: 求解第二种非线性 Volterra 积分方程:

$$x(s)=1+\int_0^s e^{-t}x^2(t)\,dt. \tag{3.25}$$

在Maple下, 输入以下命令:

```
expr4:=x(s)=1+int(exp(-t)*(x(t))^2,t=0..s);
mainproc(expr4,10);
```

即可得:

The equation can be converted into the following algebraic equations:

$$a_0-1=0$$
$$a_1-{a_0}^2=0$$

$$a_2 - a_0a_1 + \tfrac{1}{2}\, a_0{}^2 = 0$$

$$a_3 + \tfrac{2}{3}\, a_0a_1 - \tfrac{1}{6}\, a_0{}^2 - \tfrac{2}{3}\, a_0a_2 - \tfrac{1}{3}\, a_1{}^2 = 0$$

$$a_4 - \tfrac{1}{4}\, a_0a_1 - \tfrac{1}{2}\, a_1a_2 + \tfrac{1}{24}\, a_0{}^2 - \tfrac{1}{2}\, a_0a_3 + \tfrac{1}{2}\, a_0a_2 + \tfrac{1}{4}\, a_1{}^2 = 0$$

$$a_5 - \tfrac{1}{120}\, a_0{}^2 + \tfrac{1}{15}\, a_0a_1 - \tfrac{2}{5}\, a_1a_3 - \tfrac{1}{5}\, a_0a_2 - \tfrac{2}{5}\, a_0a_4 - \tfrac{1}{5}\, a_2{}^2 - \tfrac{1}{10}\, a_1{}^2 + \tfrac{2}{5}\, a_0a_3 + \tfrac{2}{5}\, a_1a_2$$
$$= 0$$

$$a_6 + \tfrac{1}{3}\, a_1a_3 + \tfrac{1}{720}\, a_0{}^2 - \tfrac{1}{72}\, a_0a_1 - 1\tfrac{1}{6}\, a_1a_2 - \tfrac{1}{6}\, a_0a_3 + \tfrac{1}{3}\, a_0a_4 - \tfrac{1}{3}\, a_2a_3 + \tfrac{1}{18}\, a_0a_2 + \tfrac{1}{6}\, a_2{}^2$$
$$+\tfrac{1}{36}\, a_1{}^2 - \tfrac{1}{3}\, a_0a_5 - \tfrac{1}{3}\, a_1a_4 = 0$$

$$a_7 + \tfrac{2}{7}\, a_0a_5 - \tfrac{2}{7}\, a_0a_6 + \tfrac{1}{420}\, a_0a_1 - \tfrac{2}{7}\, a_1a_5 + \tfrac{1}{21}\, a_1a_2 - \tfrac{1}{5040}\, a_0{}^2 + \tfrac{2}{7}\, a_1a_4 + \tfrac{1}{21}\, a_0a_3$$
$$+\tfrac{2}{7}\, a_2a_3 - \tfrac{2}{7}\, a_2a_4 - \tfrac{1}{84}\, a_0a_2 - \tfrac{1}{7}\, a_1a_3 - \tfrac{1}{7}\, a_0a_4 - \tfrac{1}{7}\, a_3{}^2 - \tfrac{1}{14}\, a_2{}^2 - \tfrac{1}{168}\, a_1{}^2 = 0$$

$$a_8 + \tfrac{1}{480}\, a_0a_2 - \tfrac{1}{96}\, a_0a_3 - \tfrac{1}{2880}\, a_0a_1 - \tfrac{1}{8}\, a_0a_5 - \tfrac{1}{96}\, a_1a_2 + \tfrac{1}{40320}\, a_0{}^2 - \tfrac{1}{8}\, a_1a_4 + \tfrac{1}{4}\, a_1a_5$$
$$-\tfrac{1}{4}\, a_1a_6 - \tfrac{1}{8}\, a_2a_3 + \tfrac{1}{4}\, a_2a_4 - \tfrac{1}{4}\, a_2a_5 - \tfrac{1}{4}\, a_3a_4 + \tfrac{1}{4}\, a_0a_6 - \tfrac{1}{4}\, a_0a_7 + \tfrac{1}{24}\, a_1a_3$$
$$+\tfrac{1}{24}\, a_0a_4 + \tfrac{1}{8}\, a_3{}^2 + \tfrac{1}{48}\, a_2{}^2 + \tfrac{1}{960}\, a_1{}^2 = 0$$

$$a_9 - \tfrac{1}{362880}\, a_0{}^2 + \tfrac{1}{22680}\, a_0a_1 - \tfrac{1}{18}\, a_3{}^2 - \tfrac{1}{216}\, a_2{}^2 - \tfrac{1}{9}\, a_4{}^2 - \tfrac{1}{6480}\, a_1{}^2 + \tfrac{1}{540}\, a_1a_2 - \tfrac{1}{108}\, a_1a_3$$
$$+\tfrac{1}{27}\, a_1a_4 - \tfrac{1}{9}\, a_1a_5 + \tfrac{2}{9}\, a_1a_6 - \tfrac{2}{9}\, a_1a_7 + \tfrac{1}{27}\, a_2a_3 - \tfrac{1}{9}\, a_2a_4 + \tfrac{2}{9}\, a_2a_5 - \tfrac{2}{9}\, a_2a_6$$
$$+\tfrac{2}{9}\, a_3a_4 - \tfrac{2}{9}\, a_3a_5 - \tfrac{1}{3240}\, a_0a_2 + \tfrac{1}{540}\, a_0a_3 - \tfrac{1}{108}\, a_0a_4 + \tfrac{1}{27}\, a_0a_5 - \tfrac{1}{9}\, a_0a_6$$
$$+\tfrac{2}{9}\, a_0a_7 - \tfrac{2}{9}\, a_0a_8 = 0$$

$$a_{10} + \tfrac{1}{3628800}\, a_0{}^2 - \tfrac{1}{201600}\, a_0a_1 - \tfrac{1}{5}\, a_0a_9 + \tfrac{1}{60}\, a_3{}^2 + \tfrac{1}{1200}\, a_2{}^2 + \tfrac{1}{10}\, a_4{}^2 + \tfrac{1}{50400}\, a_1{}^2$$
$$-\tfrac{1}{3600}\, a_1a_2 + \tfrac{1}{600}\, a_1a_3 - \tfrac{1}{120}\, a_1a_4 + \tfrac{1}{30}\, a_1a_5 - \tfrac{1}{10}\, a_1a_6 + \tfrac{1}{5}\, a_1a_7 - \tfrac{1}{5}\, a_1a_8$$
$$-\tfrac{1}{120}\, a_2a_3 + \tfrac{1}{30}\, a_2a_4 - \tfrac{1}{10}\, a_2a_5 + \tfrac{1}{5}\, a_2a_6 - \tfrac{1}{5}\, a_2a_7 - \tfrac{1}{10}\, a_3a_4 + \tfrac{1}{5}\, a_3a_5 - \tfrac{1}{5}\, a_3a_6$$
$$-\tfrac{1}{5}\, a_4a_5 + \tfrac{1}{25200}\, a_0a_2 - \tfrac{1}{3600}\, a_0a_3 + \tfrac{1}{600}\, a_0a_4 - \tfrac{1}{120}\, a_0a_5 + \tfrac{1}{30}\, a_0a_6 - \tfrac{1}{10}\, a_0a_7$$
$$+\tfrac{1}{5}\, a_0a_8 = 0$$

Solve this algebraic equations, we can get:

$$\{a_0 = 1, a_1 = 1, a_2 = \tfrac{1}{2}, a_3 = \tfrac{1}{6}, a_4 = \tfrac{1}{24}, a_5 = \tfrac{1}{120}, a_6 = \tfrac{1}{720}, a_7 = \tfrac{1}{5040}, a_8 = \tfrac{1}{40320},$$
$$a_9 = \tfrac{1}{362880}, a_{10} = \tfrac{1}{3628800}\}$$

Then the solution of the equation is as follows:

$$x(s) = 1 + s + \tfrac{1}{2}s^2 + \tfrac{1}{6}s^3 + \tfrac{1}{24}s^4 + \tfrac{1}{120}s^5 + \tfrac{1}{720}s^6 + \tfrac{1}{5040}s^7 + \tfrac{1}{40320}s^8 + \tfrac{1}{362880}s^9 + \tfrac{1}{3628800}s^{10}$$

易证, 方程 (3.25) 的解析解为:

$$x(s) = e^s = 1 + s + \frac{1}{2}s^2 + \frac{1}{6}s^3 + \frac{1}{24}s^4 + \frac{1}{120}s^5 + \frac{1}{720}s^6 + \frac{1}{5040}s^7 + \frac{1}{40320}s^8 + \frac{1}{362880}s^9 + \frac{1}{3628800}s^{10} + \cdots$$

例 3.14: 解方程:

$$x^2(s) = e^{2s} - e^s + 1 + \int_0^s e^{-t}x(t)^2 dt. \tag{3.26}$$

在Maple下输入:

```
expr5:=(x(s))^2=exp(2*s)-exp(s)+1+int(exp(-t)*(x(t))^2,
        t=0..s);
mainproc(expr5,10);
```

即得求解过程为:

```
The equation can be converted into the following algebraic
    equations:
```

$$a_0{}^2 - 1 = 0$$

$$2\,a_0a_1 + a_0{}^2 - 3 = 0$$

$$2\,a_0a_2 + a_1{}^2 - \frac{1}{2}\,a_0{}^2 + a_0a_1 - \frac{5}{2} = 0$$

$$2\,a_1a_2 + 2\,a_0a_3 + \frac{2}{3}\,a_0a_2 + \frac{1}{6}\,a_0{}^2 - \frac{2}{3}\,a_0a_1 + \frac{1}{3}\,a_1{}^2 - \frac{3}{2} = 0$$

$$2\,a_0a_4 + 2\,a_1a_3 + a_2{}^2 + \frac{1}{2}\,a_0a_3 - \frac{1}{2}\,a_0a_2 + \frac{1}{2}\,a_1a_2 - \frac{1}{24}\,a_0{}^2 - \frac{1}{4}\,a_1{}^2 + \frac{1}{4}\,a_0a_1 - \frac{17}{24} = 0$$

$$2\,a_0a_5 + \frac{1}{120}\,a_0{}^2 + \frac{1}{10}\,a_1{}^2 + \frac{1}{5}\,a_2{}^2 + \frac{2}{5}\,a_0a_4 - \frac{2}{5}\,a_0a_3 - \frac{2}{5}\,a_1a_2 - \frac{1}{15}\,a_0a_1 + \frac{1}{5}\,a_0a_2 + \frac{2}{5}\,a_1a_3 + 2\,a_1a_4 + 2\,a_2a_3 - \frac{11}{40} = 0$$

$$2\,a_0a_6 + 2\,a_1a_5 + a_3{}^2 + 2\,a_2a_4 - \frac{1}{720}\,a_0{}^2 + \frac{1}{3}\,a_2a_3 - \frac{1}{3}\,a_1a_3 - \frac{1}{36}\,a_1{}^2 + \frac{1}{6}\,a_1a_2 + \frac{1}{3}\,a_1a_4 + \frac{1}{6}\,a_0a_3 - \frac{1}{18}\,a_0a_2 - \frac{1}{3}\,a_0a_4 + \frac{1}{3}\,a_0a_5 - \frac{1}{6}\,a_2{}^2 + \frac{1}{72}\,a_0a_1 - \frac{13}{144} = 0$$

$$2\,a_0a_7 + 2\,a_1a_6 + 2\,a_3a_4 + 2\,a_2a_5 - \frac{2}{7}\,a_2a_3 + \frac{2}{7}\,a_1a_5 + \frac{1}{168}\,a_1{}^2 + \frac{1}{7}\,a_0a_4 + \frac{1}{7}\,a_1a_3 - \frac{1}{420}\,a_0a_1 + \frac{2}{7}\,a_2a_4 + \frac{1}{7}\,a_3{}^2 - \frac{2}{7}\,a_0a_5 + \frac{2}{7}\,a_0a_6 + \frac{1}{14}\,a_2{}^2 - \frac{1}{21}\,a_0a_3 - \frac{1}{21}\,a_1a_2 - \frac{2}{7}\,a_1a_4 + \frac{1}{84}\,a_0a_2 + \frac{1}{5040}\,a_0{}^2 - \frac{43}{1680} = 0$$

$$2\,a_0a_8 - \frac{1}{40320}\,a_0{}^2 + \frac{1}{8}\,a_1a_4 + \frac{1}{8}\,a_0a_5 + \frac{1}{96}\,a_1a_2 - \frac{1}{960}\,a_1{}^2 - \frac{1}{48}\,a_2{}^2 - \frac{1}{8}\,a_3{}^2 + a_4{}^2 - \frac{1}{24}\,a_1a_3 - \frac{1}{4}\,a_1a_5 + \frac{1}{4}\,a_1a_6 + 2\,a_1a_7 + \frac{1}{8}\,a_2a_3 - \frac{1}{4}\,a_2a_4 + \frac{1}{4}\,a_2a_5 + \frac{1}{4}\,a_3a_4$$

$$+2\,a_3a_5+\tfrac{1}{2880}\,a_0a_1-\tfrac{1}{480}\,a_0a_2+\tfrac{1}{96}\,a_0a_3-\tfrac{1}{24}\,a_0a_4-\tfrac{1}{4}\,a_0a_6+\tfrac{1}{4}\,a_0a_7$$
$$+2\,a_2a_6-\tfrac{257}{40320}=0$$

$$2\,a_0a_9+\tfrac{1}{362880}\,a_0{}^2-\tfrac{1}{27}\,a_1a_4-\tfrac{1}{27}\,a_0a_5-\tfrac{1}{540}\,a_1a_2+\tfrac{1}{6480}\,a_1{}^2+\tfrac{1}{216}\,a_2{}^2+\tfrac{1}{18}\,a_3{}^2$$
$$+\tfrac{1}{9}\,a_4{}^2+\tfrac{1}{108}\,a_1a_3+\tfrac{1}{9}\,a_1a_5-\tfrac{2}{9}\,a_1a_6+\tfrac{2}{9}\,a_1a_7+2\,a_1a_8-\tfrac{1}{27}\,a_2a_3+\tfrac{1}{9}\,a_2a_4$$
$$-\tfrac{2}{9}\,a_2a_5-\tfrac{2}{9}\,a_3a_4+\tfrac{2}{9}\,a_3a_5+2\,a_3a_6+2\,a_4a_5-\tfrac{1}{22680}\,a_0a_1+\tfrac{1}{3240}\,a_0a_2$$
$$-\tfrac{1}{540}\,a_0a_3+\tfrac{1}{108}\,a_0a_4+\tfrac{1}{9}\,a_0a_6-\tfrac{2}{9}\,a_0a_7+\tfrac{2}{9}\,a_0a_8+\tfrac{2}{9}\,a_2a_6$$
$$+2\,a_2a_7-\tfrac{19}{13440}=0$$

$$2\,a_0a_{10}-\tfrac{1}{3628800}\,a_0{}^2+\tfrac{1}{120}\,a_1a_4+\tfrac{1}{120}\,a_0a_5+\tfrac{1}{3600}\,a_1a_2-\tfrac{1}{50400}\,a_1{}^2-\tfrac{1}{1200}\,a_2{}^2-\tfrac{1}{60}\,a_3{}^2$$
$$-\tfrac{1}{10}\,a_4{}^2+a_5{}^2-\tfrac{1}{600}\,a_1a_3-\tfrac{1}{30}\,a_1a_5+\tfrac{1}{10}\,a_1a_6-\tfrac{1}{5}\,a_1a_7+\tfrac{1}{5}\,a_1a_8$$
$$+2\,a_1a_9+\tfrac{1}{120}\,a_2a_3-\tfrac{1}{30}\,a_2a_4+\tfrac{1}{10}\,a_2a_5+\tfrac{1}{10}\,a_3a_4-\tfrac{1}{5}\,a_3a_5+\tfrac{1}{5}\,a_3a_6$$
$$+2\,a_3a_7+\tfrac{1}{5}\,a_4a_5+2\,a_4a_6+\tfrac{1}{201600}\,a_0a_1-\tfrac{1}{25200}\,a_0a_2+\tfrac{1}{3600}\,a_0a_3$$
$$-\tfrac{1}{600}\,a_0a_4-\tfrac{1}{30}\,a_0a_6+\tfrac{1}{10}\,a_0a_7-\tfrac{1}{5}\,a_0a_8+\tfrac{1}{5}\,a_0a_9-\tfrac{1}{5}\,a_2a_6$$
$$+2\,a_2a_8+\tfrac{1}{5}\,a_2a_7-\tfrac{41}{145152}=0$$

Solve this algebraic equations, we can get:

$$\{a_0=1,a_1=1,a_2=\tfrac{1}{2},a_3=\tfrac{1}{6},a_4=\tfrac{1}{24},a_5=\tfrac{1}{120},a_6=\tfrac{1}{720},a_7=\tfrac{1}{5040},$$
$$a_8=\tfrac{1}{40320},a_9=\tfrac{1}{362880},a_{10}=\tfrac{1}{3628800}\}$$

Then the solution of the equation is as follows:

$$x(s)=1+s+\tfrac{1}{2}s^2+\tfrac{1}{6}s^3+\tfrac{1}{24}s^4+\tfrac{1}{120}s^5+\tfrac{1}{720}s^6+\tfrac{1}{5040}s^7+\tfrac{1}{40320}s^8+\tfrac{1}{362880}s^9$$
$$+\tfrac{1}{3628800}s^{10}$$

方程 (3.26) 的解析解为 $x(s)=e^s$, 显然, 机械化算法mainproc 给出的是其 10 阶 Taylor 多项式近似解.

下面, 我们再通过例子说明机械化算法mainproc对于第一种 Volterra 积分方程的求解也是有用的. 我们选取两个例子, 一个是线性方程, 一个是非线性的.

例 3.15: 求解第一种 Volterra 积分方程:

$$\int_0^s e^{s-t}x(t)dt=\cos s. \tag{3.27}$$

在Maple中输入以下命令:

```
expr6:=int(exp(s-t)*x(t),t=0..s)=cos(s);
mainproc(expr6,10);
```

即可得下述求解过程:

```
The equation can be converted into the following algebraic equations:
```

$a_0 - 1 = 0$

$a_1 + a_0 = 0$

$a_2 + \frac{1}{2}a_1 + \frac{1}{2}a_0 + \frac{1}{2} = 0$

$a_3 + \frac{1}{3}a_2 + \frac{1}{6}a_1 + \frac{1}{6}a_0 = 0$

$a_4 + \frac{1}{4}a_3 + \frac{1}{12}a_2 + \frac{1}{24}a_1 + \frac{1}{24}a_0 - \frac{1}{24} = 0$

$a_5 + \frac{1}{5}a_4 + \frac{1}{20}a_3 + \frac{1}{60}a_2 + \frac{1}{120}a_1 + \frac{1}{120}a_0 = 0$

$a_6 + \frac{1}{6}a_5 + \frac{1}{30}a_4 + \frac{1}{120}a_3 + \frac{1}{360}a_2 + \frac{1}{720}a_1 + \frac{1}{720}a_0 + \frac{1}{720} = 0$

$a_7 + \frac{1}{7}a_6 + \frac{1}{42}a_5 + \frac{1}{210}a_4 + \frac{1}{840}a_3 + \frac{1}{2520}a_2 + \frac{1}{5040}a_1 + \frac{1}{5040}a_0 = 0$

$a_8 + \frac{1}{8}a_7 + \frac{1}{56}a_6 + \frac{1}{336}a_5 + \frac{1}{1680}a_4 + \frac{1}{6720}a_3 + \frac{1}{20160}a_2 + \frac{1}{40320}a_1 + \frac{1}{40320}a_0 - \frac{1}{40320} = 0$

$a_9 + \frac{1}{9}a_8 + \frac{1}{72}a_7 + \frac{1}{504}a_6 + \frac{1}{3024}a_5 + \frac{1}{15120}a_4 + \frac{1}{60480}a_3 + \frac{1}{181440}a_2 + \frac{1}{362880}a_1$
$\quad + \frac{1}{362880}a_0 = 0$

$a_{10} + \frac{1}{10}a_9 + \frac{1}{90}a_8 + \frac{1}{720}a_7 + \frac{1}{5040}a_6 + \frac{1}{30240}a_5 + \frac{1}{151200}a_4 + \frac{1}{604800}a_3 + \frac{1}{1814400}a_2$
$\quad + \frac{1}{3628800}a_1 + \frac{1}{3628800}a_0 + \frac{1}{3628800} = 0$

```
Solve this algebraic equations, we can get:
```

$\{a_0 = 1, a_1 = -1, a_2 = -\frac{1}{2}, a_3 = \frac{1}{6}, a_4 = \frac{1}{24}, a_5 = -\frac{1}{120}, a_6 = -\frac{1}{720},$
$a_7 = \frac{1}{5040}, a_8 = \frac{1}{40320}, a_9 = -\frac{1}{362880}, a_{10} = -\frac{1}{3628800}\}$

```
Then the solution of the equation is as follows:
```

$x(s) = 1 - s - \frac{1}{2}s^2 + \frac{1}{6}s^3 + \frac{1}{24}s^4 - \frac{1}{120}s^5 - \frac{1}{720}s^6 + \frac{1}{5040}s^7 + \frac{1}{40320}s^8 - \frac{1}{362880}s^9$
$-\frac{1}{3628800}s^{10}$

方程 (3.27) 的解析解为:

$x(s) = \cos s - \sin s = 1 - s - \frac{1}{2}s^2 + \frac{1}{6}s^3 + \frac{1}{24}s^4 - \frac{1}{120}s^5 - \frac{1}{720}s^6 + \frac{1}{5040}s^7 + \frac{1}{40320}s^8$
$-\frac{1}{362880}s^9 - \frac{1}{3628800}s^{10} + \cdots$

例 3.16: 解方程:

$$\int_0^s e^{-2t}x^3(t)\,dt = e^s - 1. \tag{3.28}$$

由Maple命令:

```
expr7:=int(exp(-2*t)*(x(t))^3,t=0..s)=exp(s)-1;
mainproc(expr7, 8);
```

即可得到:

```
The equation can be converted into the following algebraic equations:
```

$$1-{a_0}^3=0$$

$$\tfrac{1}{2}-\tfrac{3}{2}\,{a_0}^2a_1+{a_0}^3=0$$

$$-{a_0}^2a_2-a_0{a_1}^2-\tfrac{2}{3}\,{a_0}^3+2\,{a_0}^2a_1+\tfrac{1}{6}=0$$

$$\tfrac{1}{24}+\tfrac{3}{2}\,{a_0}^2a_2-\tfrac{3}{4}\,{a_0}^2a_3-\tfrac{1}{4}\,{a_1}^3-\tfrac{3}{2}\,{a_0}^2a_1+\tfrac{3}{2}\,a_0{a_1}^2-\tfrac{3}{2}\,a_0a_1a_2+\tfrac{1}{3}\,{a_0}^3=0$$

$$\begin{aligned}&-\tfrac{3}{5}\,{a_0}^2a_4-\tfrac{6}{5}\,{a_0}^2a_2+\tfrac{2}{5}\,{a_1}^3-\tfrac{6}{5}\,a_0{a_1}^2-\tfrac{2}{15}\,{a_0}^3+\tfrac{12}{5}\,a_0a_1a_2-\tfrac{3}{5}\,a_0{a_2}^2-\tfrac{3}{5}\,{a_1}^2a_2\\&\quad+\tfrac{4}{5}\,{a_0}^2a_1-\tfrac{6}{5}\,a_0a_1a_3+\tfrac{6}{5}\,{a_0}^2a_3+\tfrac{1}{120}=0\end{aligned}$$

$$\begin{aligned}&-{a_0}^2a_3-\tfrac{1}{3}\,{a_1}^3+a_0{a_2}^2-\tfrac{1}{2}\,{a_0}^2a_5+{a_0}^2a_4-\tfrac{1}{3}\,{a_0}^2a_1-a_0a_2a_3+\tfrac{2}{45}\,{a_0}^3-\tfrac{1}{2}\,a_1{a_2}^2\\&\quad-2\,a_0a_1a_2+{a_1}^2a_2+\tfrac{2}{3}\,a_0{a_1}^2-a_0a_1a_4+2\,a_0a_1a_3-\tfrac{1}{2}\,{a_1}^2a_3+\tfrac{2}{3}\,{a_0}^2a_2+\tfrac{1}{720}=0\end{aligned}$$

$$\begin{aligned}&-\tfrac{1}{7}\,{a_2}^3-\tfrac{6}{7}\,a_0a_2a_4-\tfrac{6}{7}\,a_1a_2a_3-\tfrac{6}{7}\,{a_0}^2a_4-\tfrac{2}{7}\,a_0{a_1}^2-\tfrac{6}{7}\,a_0a_1a_5-\tfrac{4}{315}\,{a_0}^3+\tfrac{12}{7}\,a_0a_2a_3\\&\quad-\tfrac{3}{7}\,{a_1}^2a_4+\tfrac{6}{7}\,{a_1}^2a_3-\tfrac{3}{7}\,a_0{a_3}^2+\tfrac{6}{7}\,a_1{a_2}^2+\tfrac{4}{7}\,{a_0}^2a_3-\tfrac{2}{7}\,{a_0}^2a_2-\tfrac{3}{7}\,{a_0}^2a_6\\&\quad-\tfrac{6}{7}\,{a_1}^2a_2+\tfrac{12}{7}\,a_0a_1a_4+\tfrac{4}{21}\,{a_1}^3+\tfrac{8}{7}\,a_0a_1a_2-\tfrac{6}{7}\,a_0{a_2}^2+\tfrac{6}{7}\,{a_0}^2a_5-\tfrac{12}{7}\,a_0a_1a_3\\&\quad+\tfrac{4}{35}\,{a_0}^2a_1+\tfrac{1}{5040}=0\end{aligned}$$

$$\begin{aligned}&\tfrac{3}{2}\,a_0a_1a_5+\tfrac{3}{2}\,a_0a_2a_4-\tfrac{3}{2}\,a_0a_2a_3-\tfrac{3}{4}\,a_0a_3a_4-\tfrac{1}{2}\,a_0a_1a_2-\tfrac{3}{4}\,a_0a_2a_5-\tfrac{3}{2}\,a_0a_1a_4+a_0a_1a_3\\&\quad+\tfrac{1}{315}\,{a_0}^3-\tfrac{3}{4}\,a_0a_1a_6-\tfrac{1}{12}\,{a_1}^3+\tfrac{1}{4}\,{a_2}^3+\tfrac{3}{2}\,a_1a_2a_3-\tfrac{3}{4}\,a_1a_2a_4+\tfrac{1}{10}\,a_0{a_1}^2\\&\quad+\tfrac{1}{2}\,a_0{a_2}^2+\tfrac{3}{4}\,a_0{a_3}^2-\tfrac{1}{4}\,{a_0}^2a_3-\tfrac{1}{30}\,{a_0}^2a_1+\tfrac{1}{10}\,{a_0}^2a_2-\tfrac{3}{4}\,{a_0}^2a_5+\tfrac{3}{4}\,{a_0}^2a_6\\&\quad-\tfrac{3}{8}\,{a_0}^2a_7+\tfrac{1}{2}\,{a_0}^2a_4+\tfrac{1}{2}\,{a_1}^2a_2-\tfrac{3}{4}\,{a_1}^2a_3+\tfrac{3}{4}\,{a_1}^2a_4-\tfrac{3}{8}\,{a_1}^2a_5-\tfrac{3}{4}\,a_1{a_2}^2\\&\quad-\tfrac{3}{8}\,a_1{a_3}^2-\tfrac{3}{8}\,{a_2}^2a_3+\tfrac{1}{40320}=0\end{aligned}$$

$$\begin{aligned}&-\tfrac{2}{3}\,a_0a_2a_6-\tfrac{4}{3}\,a_0a_1a_5-\tfrac{4}{3}\,a_0a_2a_4-\tfrac{2}{3}\,a_0a_3a_5+\tfrac{8}{9}\,a_0a_2a_3+\tfrac{4}{3}\,a_0a_3a_4-\tfrac{2}{3}\,a_0a_1a_7\\&\quad+\tfrac{8}{45}\,a_0a_1a_2+\tfrac{4}{3}\,a_0a_2a_5+\tfrac{8}{9}\,a_0a_1a_4-\tfrac{4}{9}\,a_0a_1a_3-\tfrac{2}{2835}\,a_0^3+\tfrac{4}{3}\,a_0a_1a_6+\tfrac{4}{135}\,{a_1}^3\\&\quad-\tfrac{2}{9}\,{a_2}^3-\tfrac{4}{3}\,a_1a_2a_3+\tfrac{4}{3}\,a_1a_2a_4-\tfrac{2}{3}\,a_1a_2a_5-\tfrac{2}{3}\,a_1a_3a_4-\tfrac{4}{135}\,a_0{a_1}^2-\tfrac{2}{9}\,a_0{a_2}^2\\&\quad-\tfrac{2}{3}\,a_0{a_3}^2+\tfrac{4}{45}\,{a_0}^2a_3+\tfrac{8}{945}\,{a_0}^2a_1-\tfrac{4}{135}\,{a_0}^2a_2+\tfrac{4}{9}\,{a_0}^2a_5-\tfrac{2}{3}\,{a_0}^2a_6+\tfrac{2}{3}\,{a_0}^2a_7\\&\quad-\tfrac{1}{3}\,{a_0}^2a_8-\tfrac{2}{9}\,{a_0}^2a_4-\tfrac{2}{9}\,{a_1}^2a_2+\tfrac{4}{9}\,{a_1}^2a_3-\tfrac{2}{3}\,{a_1}^2a_4+\tfrac{2}{3}\,{a_1}^2a_5-\tfrac{1}{3}\,{a_1}^2a_6\\&\quad+\tfrac{4}{9}\,a_1{a_2}^2+\tfrac{2}{3}\,a_1{a_3}^2+\tfrac{2}{3}\,{a_2}^2a_3-\tfrac{1}{3}\,{a_2}^2a_4-\tfrac{1}{3}\,a_2{a_3}^2-\tfrac{1}{3}\,a_0{a_4}^2+\tfrac{1}{362880}=0\end{aligned}$$

```
Solve this algebraic equations, we can get:
```

$$\left\{a_0=1,a_1=1,a_2=\tfrac{1}{2},a_3=\tfrac{1}{6},a_4=\tfrac{1}{24},a_5=\tfrac{1}{120},a_6=\tfrac{1}{720},a_7=\tfrac{1}{5040},a_8=\tfrac{1}{40320}\right\}$$

```
Then the solution of the equation is as follows:
```

$$x(s)=1+s+\frac{1}{2}s^2+\frac{1}{6}s^3+\frac{1}{24}s^4+\frac{1}{120}s^5+\frac{1}{720}s^6+\frac{1}{5040}s^7+\frac{1}{40320}s^8$$

容易验证 (3.28) 的解析解为:

$$x(s) = e^s = 1 + s + \tfrac{1}{2}s^2 + \tfrac{1}{6}s^3 + \tfrac{1}{24}s^4 + \tfrac{1}{120}s^5 + \tfrac{1}{720}s^6 + \tfrac{1}{5040}s^7 + \tfrac{1}{40320}s^8 + \cdots$$

最后我们给出一个利用机械化算法mainproc求解积分 — 微分方程的例子. 关于更复杂的积分 — 微分方程的求解问题我们将在下一章讨论.

例 3.17: 求解:

$$x''(s) = 1 + se^s - \int_0^s e^{s-t}\, x(t)\, dt. \tag{3.29}$$

在Maple中输入:

```
expr8:=diff(x(s),s$2)=1+s*exp(s)-int(exp(s-t)*x(t),t=0..s);
mainproc(expr8,8);
```

即可得到以下结果:

The equation can be converted into the following algebraic equations:

$a_0 = 0$

$a_1 + a_0 - 1 = 0$

$a_2 + \tfrac{1}{2}a_1 + \tfrac{1}{2}a_0 - 1 = 0$

$a_3 + \tfrac{1}{3}a_2 + \tfrac{1}{6}a_1 + \tfrac{1}{6}a_0 - \tfrac{1}{2} = 0$

$a_4 + \tfrac{1}{4}a_3 + \tfrac{1}{12}a_2 + \tfrac{1}{24}a_1 + \tfrac{1}{24}a_0 - \tfrac{1}{6} = 0$

$a_5 + \tfrac{1}{5}a_4 + \tfrac{1}{20}a_3 + \tfrac{1}{60}a_2 + \tfrac{1}{120}a_1 + \tfrac{1}{120}a_0 - \tfrac{1}{24} = 0$

$a_6 + \tfrac{1}{6}a_5 + \tfrac{1}{30}a_4 + \tfrac{1}{120}a_3 + \tfrac{1}{360}a_2 + \tfrac{1}{720}a_1 + \tfrac{1}{720}a_0 - \tfrac{1}{120} = 0$

$a_7 + \tfrac{1}{7}a_6 + \tfrac{1}{42}a_5 + \tfrac{1}{210}a_4 + \tfrac{1}{840}a_3 + \tfrac{1}{2520}a_2 + \tfrac{1}{5040}a_1 + \tfrac{1}{5040}a_0 - \tfrac{1}{720} = 0$

$a_8 + \tfrac{1}{8}a_7 + \tfrac{1}{56}a_6 + \tfrac{1}{336}a_5 + \tfrac{1}{1680}a_4 + \tfrac{1}{6720}a_3 + \tfrac{1}{20160}a_2 + \tfrac{1}{40320}a_1 + \tfrac{1}{40320}a_0 - \tfrac{1}{5040} = 0$

Solve this algebraic equations, we can get:

$\{a_0 = 0, a_1 = 1, a_2 = \tfrac{1}{2}, a_3 = \tfrac{1}{6}, a_4 = \tfrac{1}{24}, a_5 = \tfrac{1}{120}, a_6 = \tfrac{1}{720}, a_7 = \tfrac{1}{5040}, a_8 = \tfrac{1}{40320}\}$

Then the solution of the equation is as follows:

$$x(s) = s + \frac{1}{2}s^2 + \frac{1}{6}s^3 + \frac{1}{24}s^4 + \frac{1}{120}s^5 + \frac{1}{720}s^6 + \frac{1}{5040}s^7 + \frac{1}{40320}s^8$$

容易验证, 方程 (3.29) 的解为:

$$x(s) = e^s - 1 = s + \frac{1}{2}s^2 + \frac{1}{6}s^3 + \frac{1}{24}s^4 + \frac{1}{120}s^5 + \frac{1}{720}s^6 + \frac{1}{5040}s^7 + \frac{1}{40320}s^8 + \cdots$$

3.3 Volterra 积分方程组求解

前面讨论了 Volterra 积分方程求解及其机械化问题. 但值得注意的是, Volterra 积分方程组求解的文献并不多见. E.Babolian [3], H.S.Goghary [41] 各自讨论了利用 Adomain 分解法求解线性积分方程组的问题. L.M.Delves 和 J.L.Mohamed 在文献 [26] 中, 介绍了两个方程组成的方程组的求解过程. 在文献 [15, 61, 114] 中, 作者简单介绍了求解 Volterra 积分方程的问题, 也提到了使用迭代法可以求解, 但并未对如何求解给出详细的介绍.

本节讨论 Volterra 积分方程组的求解问题, 并在此基础上借助Maple建立机械化算法.

3.3.1 基本算法

考虑 Volterra 积分方程组:

$$\begin{cases} x_1(s) - \lambda \int_a^s k_{11}(s,t)x_1(t)dt - \lambda \int_a^s k_{12}(s,t)x_2(t)dt \\ \qquad - \cdots - \lambda \int_a^s k_{1n}(s,t)x_n(t)dt = f_1(s), \\ x_2(s) - \lambda \int_a^s k_{21}(s,t)x_1(t)dt - \lambda \int_a^s k_{22}(s,t)x_2(t)dt \\ \qquad - \cdots - \lambda \int_a^s k_{2n}(s,t)x_n(t)dt = f_2(s), \\ \cdots\cdots \\ x_n(s) - \lambda \int_a^s k_{n1}(s,t)x_1(t)dt - \lambda \int_a^s k_{n2}(s,t)x_2(t)dt \\ \qquad - \cdots - \lambda \int_a^s k_{nn}(s,t)x_n(t)dt = f_n(s). \end{cases} \tag{3.30}$$

这里, 函数 $f_i(s) \in L^2[a,b]$ 是已知函数, $k_{ij}(s,t)$ $(i,j=1,2,\cdots,n)$ 是 L^2- 核函数, λ 是参数, 而 $x_i(s)$ $(i=1,2,\cdots,n)$ 是要求解的函数.

将方程组 (3.30) 重写为下述形式:

$$x_i(s) - \lambda \sum_{j=1}^{n} \int_a^s k_{ij}(s,t)x_j(t)dt = f_i(s), \quad i=1,2,\cdots,n. \tag{3.31}$$

假设

$$x_i(s) = f_i(s) + \sum_{n=1}^{\infty} \lambda^n \varphi_{in}(s), \quad i=1,2,\cdots,n. \tag{3.32}$$

其中, $\varphi_{in}(s)$ $(i=1,2,\cdots,n)$ 是要求解的函数. 将 (3.32) 代入 (3.31), 并逐项积分, 可得:

$$\sum_{n=1}^{\infty} \lambda^n \varphi_{in}(s) = \sum_{j=1}^{n} \left\{ \lambda \int_a^s k_{ij}(s,t)f_j(t)dt + \sum_{n=1}^{\infty} \lambda^{n+1} \int_a^s k_{ij}(s,t)\varphi_{in}(t)dt \right\}. \tag{3.33}$$

令 λ 的各次幂的系数为 0, 即得:

$$\begin{aligned}\varphi_{i1}(s) &= \sum_{j=1}^{n} \int_a^s k_{ij}(s,t) f_j(t) dt,\\ \varphi_{im}(s) &= \sum_{j=1}^{n} \int_a^s k_{ij}(s,t) \varphi_{jm-1}(t)(t) dt, \quad (m = 2, 3, \cdots)\end{aligned} \tag{3.34}$$

下面给出方程组 (3.30) 的求解法则:

定理 3.2: 假设积分核函数 $k_{ij}(s,t) \in L^2[a,b]$ $(i,j = 1,2,\cdots,n)$ 和 $f_i(s) \in L^2[a,b]$ 有界, 则对于任意 λ, 方程组 (3.30) 有唯一解:

$$x_i(s) = \sum_{n=0}^{\infty} \lambda^n \varphi_{in}(s), \quad i = 1, 2, \cdots, n. \tag{3.35}$$

其中, $\varphi_{i0}(s) = f_i(s)$, $\varphi_{in}(s)$ $(n = 1, 2, \cdots)$ 由公式 (3.34) 定义.

证明: 由 $k_{ij}(s,t)$ $(i,j = 1,2,\cdots,n)$ 和 $f_i(s)$ $(i = 1,2,\cdots,n)$ 有界, 可假设

$$|k_{ij}(s,t)| \leqslant M, \quad |f_i(s)| \leqslant m.$$

接下来, 我们依次计算 $\varphi_{in}(s)$ $(n = 0, 1, 2, \cdots)$.

$$\begin{aligned}
&|\varphi_{i0}(s)| = |f_i(s)| \leqslant m,\\
&|\varphi_{i1}(s)| = |\sum_{j=1}^{n} \int_a^s k_{ij}(s,t) f_j(t) dt| \leqslant \sum_{j=1}^{n} \int_a^s |k_{ij}(s,t)| \cdot |f_j(t)| dt \leqslant nmM(s-a),\\
&|\varphi_{i2}(s)| = |\sum_{j=1}^{n} \int_a^s k_{ij}(s,t) \varphi_{j1}(t) dt| \leqslant \sum_{j=1}^{n} \int_a^s |k_{ij}(s,t)| \cdot |\varphi_{j1}(t)| dt \leqslant n^2 m M^2 \int_a^s (t-a) dt\\
&\qquad = m \cdot \tfrac{(nM(s-a))^2}{2!},\\
&\cdots\cdots\\
&|\varphi_{ip}(s)| = |\sum_{j=1}^{n} \int_a^s k_{ij}(s,t) \varphi_{jn-1}(t) dt| \leqslant \sum_{j=1}^{n} \int_a^s |k_{ij}(s,t)| \cdot |\varphi_{jn-1}(t)| dt\\
&\qquad \leqslant nM \int_a^s \tfrac{(nM(t-a))^{n-1}}{(n-1)!} dt = m \cdot \tfrac{(nM(s-a))^n}{n!},\\
&\cdots\cdots
\end{aligned} \tag{3.36}$$

由于 $a \leqslant s \leqslant b$, 从而 (3.35) 的通项满足

$$|\lambda^n \varphi_{in}(s)| \leqslant m \cdot \frac{(|\lambda| nM(s-a))^n}{n!} \tag{3.37}$$

由此, 对任意的 λ, 正项级数 $\{m\cdot\frac{(|\lambda|nM(s-a))^n}{n!}\}$ 收敛. 所以, (3.35) 在 $[a,b^*]$ $(b^*<b)$ 上一致收敛.

注意到迭代过程以及关系式 (3.32)—(3.34), 我们知道和函数

$$x_i(s)=\sum_{n=0}^{\infty}\lambda^n\varphi_{in}(s),\quad i=1,2,\cdots,n$$

在区间 $[a,b]$ 上连续并且满足方程组 (3.30). 证毕.

然而, 在实际计算中, 对于迭代过程 (3.34), 我们并不能通过有限项结果 $\{\varphi_{in}(s)\}$ 得到其通项公式 $\varphi_{in}(s)$. 于是, 我们可取下面我们给出截断误差的估计式.

由

$$\left|\varphi_{in}(s)\right|<m\cdot\frac{(nM(s-a))^n}{n!}\tag{3.38}$$

可得余项的估计式为:

$$\begin{aligned}&R_{n+1}(s)=|\lambda^{n+1}\varphi_{in+1}(s)+\lambda^{n+2}\varphi_{in+2}(s)+\lambda^{n+3}\varphi_{in+3}(s)+\cdots|\\&<|\lambda|^{n+1}m\cdot\tfrac{(nM(s-a))^{n+1}}{(n+1)!}+|\lambda|^{n+2}m\cdot\tfrac{(nM(s-a))^{n+2}}{(n+2)!}+|\lambda|^{n+3}m\cdot\tfrac{(nM(s-a))^{n+3}}{(n+3)!}+\cdots\\&<|\lambda|^{n+1}m\cdot\tfrac{(nM(s-a))^{n+1}}{(n+1)!}+|\lambda|^{n+2}m\cdot\tfrac{(nM(s-a))^{n+2}}{(n+1)!}+|\lambda|^{n+3}m\cdot\tfrac{(nM(s-a))^{n+3}}{(n+1)!}+\cdots\\&=\tfrac{|\lambda|^{n+1}m(nM(s-a))^{n+1}}{(n+1)!(1-|\lambda|nM(s-a))}.\end{aligned}\tag{3.39}$$

为了更清楚地理解迭代法求解 Volterra 积分方程组, 我们给出一个简单的例子.

例 3.18: 求解方程:

$$\begin{aligned}&x_1(s)-\textstyle\int_0^s x_2(t)dt=s,\\&x_2(s)-\textstyle\int_0^s x_1(t)dt=1.\end{aligned}\tag{3.40}$$

这里,

$$k_{11}(s,t)=0,\ k_{12}(s,t)=1,\ k_{21}(s,t)=1,\ k_{22}(s,t)=0,\ f_1(s)=s,\ f_2(s)=1,\ \lambda=1.$$

由公式 (3.34), 我们可以得到:

$$
\begin{aligned}
&\varphi_{11}(s)=s, \quad \varphi_{21}(s)=\tfrac{s^2}{2},\\
&\varphi_{12}(s)=\tfrac{s^3}{6}, \quad \varphi_{22}(s)=\tfrac{s^2}{2},\\
&\varphi_{13}(s)=\tfrac{s^3}{6}, \quad \varphi_{23}(s)=\tfrac{s^4}{24},\\
&\varphi_{14}(s)=\tfrac{s^5}{120}, \quad \varphi_{24}(s)=\tfrac{s^4}{24},\\
&\varphi_{15}(s)=\tfrac{s^5}{120}, \quad \varphi_{25}(s)=\tfrac{s^6}{720},\\
&\varphi_{16}(s)=\tfrac{s^7}{5040}, \quad \varphi_{26}(s)=\tfrac{s^6}{720},\\
&\varphi_{17}(s)=\tfrac{s^7}{5040}, \quad \varphi_{27}(s)=\tfrac{s^8}{40320},\\
&\varphi_{18}(s)=\tfrac{s^9}{362880}, \quad \varphi_{28}(s)=\tfrac{s^8}{40320},\\
&\varphi_{19}(s)=\tfrac{s^9}{362880}, \quad \varphi_{29}(s)=\tfrac{s^{10}}{362880}.
\end{aligned}
$$

由公式 (3.35), 可得方程 (3.40) 的 9 阶截断近似解为:

$$
\begin{aligned}
x_1(s)&=f_1(s)+\sum_{j=1}^{j}\varphi_{1j}(s)=2\,s+\tfrac{1}{3}\,s^3+\tfrac{1}{60}\,s^5+\tfrac{1}{2520}\,s^7+\tfrac{1}{181440}\,s^9+\cdots,\\
x_2(s)&=f_2(s)+\sum_{j=1}^{j}\varphi_{2j}(s)=1+s^2+\tfrac{1}{12}\,s^4+\tfrac{1}{360}\,s^6+\tfrac{1}{20160}\,s^8+\cdots.
\end{aligned}
\tag{3.41}
$$

另一方面, 我们可以得到 (3.40) 的解析解为:

$$
x_1(s)=e^s-e^{-s}, \quad x_2(s)=-1+e^s+e^{-s}. \tag{3.42}
$$

显然, $x_1(s)$ 和 $x_2(s)$ 的 Taylor 展开式如下所示:

$$
\begin{aligned}
x_1(s)&=e^s-e^{-s}=2\,s+\tfrac{1}{3}\,s^3+\tfrac{1}{60}\,s^5+\tfrac{1}{2520}\,s^7+\cdots,\\
x_2(s)&=-1+e^s+e^{-s}=1+s^2+\tfrac{1}{12}\,s^4+\tfrac{1}{360}\,s^6+\cdots.
\end{aligned}
\tag{3.43}
$$

比较 (3.41) 和 (3.43) 可知, 迭代法是求解 Volterra 积分方程组 (3.30) 的一种有效的方法. 另外, 通过例 3.18 的求解过程, 我们可以看到, Volterra 积分方程组求解的计算量很大. 下节, 将建立基于迭代法的机械化算法.

3.3.2 Volterra 积分方程组的机械化算法

我们可以应用公式 (3.32)— (3.35) 在`Maple`下求解 Volterra 积分方程组 (3.30), 其求解过程可以在`Maple`建立相应的机械化算法`VoltEqns`, 源代码列示如下:

```
VoltEqns:=proc(funSet::list,n::integer)
```

```
local xFunList,yFunList,intExprList,tempList,kList,i,j,
      Lam,Nf,temp,xList,Phi,PhiT;
xFunList:=[]:yFunList:=[]:intExprList:=[]:
kList:=[]:xList:=[]:temp:=[]:
Nf:=nops(funSet);
for i from 1 to Nf do
      tempList:=parmeint(funSet[i]):
      temp:=[op(temp),tempList[5]];
od;
Lam:=lcm(op(temp));
for i from 1 to Nf do
      tempList:=parmeint(funSet[i]):
      xFunList:=[op(xFunList),tempList[1]]:
      yFunList:=[op(yFunList),tempList[2]]:
      intExprList:=[op(intExprList),tempList[3]
            *tempList[5]/Lam]:
      if nops([op(expand(intExprList[i]))])<>nops(funSet)
          then print('Input Data Error'):  return:
      end if:
      kList:=[op(kList),tempList[4]*tempList[5]/Lam]:
end do:
lprint('Here,the parameters of the system are as follows:');
for i from 1 to nops(yFunList) do
      printf("f[%a](s)=%a,        ",i,op(i,yFunList));
od;
printf("\n");
for i from 1 to nops(op(1,kList)) do
      for j from 1 to nops(op(1,kList)) do
          printf("k[%a%a](s,t)=%a,  ",j,i,op(i,op(j,kList)))
      od;
od;
printf("\n");
lprint('and');
print(lambda=Lam);
Phi:=[subs(s=t,yFunList)];
```

```
for i from 1 to n+1 do
      PhiT:=expand(subs(seq(x[j](t)=Phi[i,j],j=1..nops
           (xFunList)),intExprList));
      Phi:=[op(Phi),subs(s=t,PhiT)];
od;
Phi:=subs(t=s,Phi);
lprint('So, we can get the approximate solution as
           follows:');
for i from 1 to nops(xFunList) do
     tempList:= add(Lam^(j-1)*Phi[j,i],j=2..n+1)+Phi[1,i]:
     tempList:=polynsort(tempList,s):
     print(x[i](s)=tempList):
od:
end proc:
```

这里, 参数funSet是要求解的方程组的信息, 而n是迭代次数. VoltEqns 中子程序parmeint 的主要作用是获取描述方程组的信息, 提取 $k_{ij}(s,t)$, $f_i(s)$, λ 及 $[a,s]$(参见 2.1.2). 而子程序polynsort 是我们建立的适用于本节问题解决的排序算法, 详细内容见 3.4 节讨论.

例如, 要求解例 3.18 即方程组 (3.40), 我们所要做的事情就是在Maple中按照要求输入方程组的信息并设定迭代次数, 如下所示:

```
expr1:=x[1](s)-int(x[2](t),t=0..s)=s;
expr2:=x[2](s)-int(x[1](t),t=0..s)=1;
VoltEqns([expr1, expr2],12);
```

然后, 机械化算法VoltEqns立即返回结果形如 (3.41).

3.3.3　应用算例

例 3.19: 求解方程组:

$$\begin{aligned} &x_1(s) - \int_0^s x_1(t)\,dt - \int_0^s x_2(t)\,dt = s, \\ &x_2(s) - \int_0^s x_1(t)\,dt - \int_0^s x_2(t)\,dt = 1. \end{aligned} \tag{3.44}$$

在Maple中, 输入以下命令:

```
expr1:=x[1](s)-int(x[1](t),t=0..s)-int(x[2](t),t=0..s)=s;
expr2:=x[2](s)-int(x[1](t),t=0..s)-int(x[2](t),t=0..s)=1;
VoltEqns([expr1,expr2],10);
```

即可得到求解结果:

```
Here,the parameters of the system are as follows:
    f[1](s)=s,  f[2](s)=1,
    k[11](s,t)=1,  k[21](s,t)=1,  k[12](s,t)=1,  k[22](s,t)=1,
and
    λ = 1
    So, we can get the approximate solution as follows:
```

$$x_1(s) = 2s + \frac{3}{2}s^2 + s^3 + \frac{1}{2}s^4 + \frac{1}{5}s^5 + \frac{1}{15}s^6 + \frac{2}{105}s^7 + \frac{1}{210}s^8 + \frac{1}{945}s^9 + \frac{1}{4725}s^{10}$$

$$x_2(s) = 1 + s + \frac{3}{2}s^2 + s^3 + \frac{1}{2}s^4 + \frac{1}{5}s^5 + \frac{1}{15}s^6 + \frac{2}{105}s^7 + \frac{1}{210}s^8 + \frac{1}{945}s^9 + \frac{1}{4725}s^{10}$$

易知, 方程 (3.44) 的解析解为:

$$x_1(s) = \frac{3}{4}e^{2s} + \frac{s}{2} - \frac{3}{4}, \quad x_2(s) = \frac{3}{4}e^{2s} - \frac{s}{2} + \frac{1}{4}.$$

其 Taylor 展开式为:

$$x_1(s) = \tfrac{3}{4}e^{2s} + \tfrac{s}{2} - \tfrac{3}{4} = 2s + \tfrac{3}{2}s^2 + s^3 + \tfrac{1}{2}s^4 + \tfrac{1}{5}s^5 + \tfrac{1}{15}s^6 + \tfrac{2}{105}s^7 + \tfrac{1}{210}s^8 + \tfrac{1}{945}s^9 + \tfrac{1}{4725}s^{10} + \cdots,$$

$$x_2(s) = \tfrac{3}{4}e^{2s} - \tfrac{s}{2} + \tfrac{1}{4} = 1 + s + \tfrac{3}{2}s^2 + s^3 + \tfrac{1}{2}s^4 + \tfrac{1}{5}s^5 + \tfrac{1}{15}s^6 + \tfrac{2}{105}s^7 + \tfrac{1}{210}s^8 + \tfrac{1}{945}s^9 + \tfrac{1}{4725}s^{10} + \cdots$$

例 3.20: 求解第二种 Volterra 积分方程组:

$$\begin{aligned} x_1(s) - \int_0^s (s-t)^2 x_1(t)\,dt + \int_0^s x_2(t)\,dt &= s^2 + 2s + 1, \\ x_2(s) - \int_0^s \frac{(s-t)}{2} x_1(t)\,dt - \int_0^s \frac{(s-t)}{2} x_2(t)\,dt &= s + 1. \end{aligned} \tag{3.45}$$

在Maple下输入:

```
expr1:=x[1](s)-int((s-t)^2*x[1](t),t=0..s)+int(x[2](t),
```

```
    t=0..s)=s^2+2*s+1;
expr2:=x[2](s)-int((s-t)/2*x[1](t),t=0..s)-int((s-t)/2
    *x[2](t),t=0..s)=s+1;
VoltEqns([expr1,expr2],6):
```

即可得机械化求解过程:

```
Here,the parameters of the system are as follows:
f[1](s)=s^2+2*s+1, f[2](s)=s+1,
k[11](s,t)=t^2-2*t*s+s^2,  k[21](s,t)=-1/2*t+1/2*s,
k[12](s,t)=-1,  k[22](s,t)=-1/2*t+1/2*s,
and
λ = 1
So, we can get the approximate solution as follows:
```

$$x_1(s)=1+s+\tfrac{1}{2}s^2+\tfrac{1}{6}s^3+\tfrac{1}{24}s^4+\tfrac{1}{120}s^5+\tfrac{1}{720}s^6+\tfrac{1}{5040}s^7+\tfrac{1}{40320}s^8+\tfrac{1}{362880}s^9 +\tfrac{1}{4147200}s^{10}+\tfrac{23}{638668800}s^{11}+\tfrac{47}{5109350400}s^{12}+\tfrac{19}{14233190400}s^{13}-\tfrac{1}{43589145600}s^{14}$$

$$x_2(s)=1+s+\tfrac{1}{2}s^2+\tfrac{1}{6}s^3+\tfrac{1}{24}s^4+\tfrac{1}{120}s^5+\tfrac{1}{720}s^6+\tfrac{1}{5040}s^7+\tfrac{1}{40320}s^8+\tfrac{1}{362880}s^9 +\tfrac{1}{3628800}s^{10}+\tfrac{1}{30412800}s^{11}+\tfrac{1}{306561024}s^{12}-\tfrac{19}{132843110400}s^{13}-\tfrac{1}{11480268800}s^{14}$$

易知, 方程 (3.45) 的解析解为:

$$x_1(s)=x_2(s)=e^s=1+s+\tfrac{1}{2}s^2+\tfrac{1}{6}s^3+\tfrac{1}{24}s^4+\tfrac{1}{120}s^5+\tfrac{1}{720}s^6+\tfrac{1}{5040}s^7+\tfrac{1}{40320}s^8 +\tfrac{1}{362880}s^9+\tfrac{1}{3628800}s^{10}+\cdots$$

例 3.21: 求解第一种 Volterra 积分方程组:

$$\begin{aligned} &\int_0^s (t-s)\,x_1(t)\,dt+\int_0^s x_2(t)\,dt=s,\\ &\int_0^s x_1(t)\,dt-\int_0^s (s-t)\,x_2(t)\,dt=\frac{s^2}{2}. \end{aligned} \tag{3.46}$$

在Maple中输入:

```
expr1:=int((t-s)*x[1](t),t=0..s)+int(x[2](t),t=0..s)=s;
expr2:=int(x[1](t),t=0..s)-int((s-t)*x[2](t),t=0..s)
    =s*s/2;
VoltEqns([expr1,expr2],15);
```

即得求解过程为:

```
Here,the parameters of the system are as follows:
f[1](s)=s,  f[2](s)=1/2*s^2,
k[11](s,t)=s-t,  k[21](s,t)=-1,
k[12](s,t)=-1,  k[22](s,t)=s-t,
and
λ = 1
So, we can get the approximate solution as follows:
```

$x_1(s) = 2s + \frac{1}{3}s^3 + \frac{1}{60}s^5 + \frac{1}{2520}s^7 + \frac{1}{181440}s^9 + \frac{1}{19958400}s^{11} + \frac{1}{3113510400}s^{13}$

$x_2(s) = 1+s^2+\frac{1}{12}s^4+\frac{1}{360}s^6+\frac{1}{20160}s^8+\frac{1}{1814400}s^{10}+\frac{1}{239500800}s^{12}+\frac{1}{43589145600}s^{14}$

容易验证, 方程组 (3.46) 的解析解为:

$$\begin{aligned} x_1(s) &= e^s - e^{-s} \\ &= 2s + \tfrac{1}{3}s^3 + \tfrac{1}{60}s^5 + \tfrac{1}{2520}s^7 + \tfrac{1}{181440}s^9 + \tfrac{1}{19958400}s^{11} + \tfrac{1}{3113510400}s^{13} + \cdots, \end{aligned}$$

$$\begin{aligned} x_2(s) &= e^s + e^{-s} - 1 \\ &= 1 + s^2 + \tfrac{1}{12}s^4 + \tfrac{1}{360}s^6 + \tfrac{1}{20160}s^8 + \tfrac{1}{1814400}s^{10} + \tfrac{1}{239500800}s^{12} + \tfrac{1}{43589145600}s^{14} \\ &\quad + \cdots \end{aligned}$$

3.4 本章小结

本章研究了 Volterra 积分方程的 Neumann 级数与 Taylor 级数求解法, 以及求解 Volterra 方程组的迭代法, 在此基础上, 在Maple平台建立了相应的机械化算法, 利用这些算法, 可以获得此类积分方程 (组) 的解析解或者近似解. 应用算例表明, 上述算法是求解 Volterra 积分方程 (组) 的高效算法之一. 这将为 Volterra 积分方程 (组) 求解提供参考.

在算法VoltEqns设计过程中, 我们发现Maple下用于排序的命令sort的一个漏洞, 即sort只能将代数表达式按递减的顺序排列, 而我们无法得到升序的排列, 虽然, Maple下sort的帮助文件说明此命令能够完成升序的任务, 但我们尝试了所有的方法均无法实现我们想要的结果. 而在 Volterra 积分方程组求解过程中输出结果的升序排列对我们判断迭代次数 n 的增加产生的变化是重要的.

事实上, 在求解 Volterra 积分方程组时, 要想获得满意的结果, 我们需要通过数学实验的方法比对分析不同的 n 所得的结果, 然后确定合适的 n 的大小.

由于这个原因, 我们建立了对函数表达式进行升序排列的算法polynsort, 这是机械化算法VoltEqns中的一个子程序. 其源码如下所示:

```
polynsort:=proc(expr,x::symbol)
local tempList,temp,tempStr,minPos,xPos,exprList,resList, i,j,n:
```

```
templist:=[op(expr)]:
exprList:=[]:resList:=[]:
for i from 1 to nops(tempList) do
      tempStr:=convert(tempList[i],string):
      xPos:=searchtext(cat(x,'^'),tempStr):
      if xPos=0 then
          if searchtext(x,tempStr)<>0 then
             exprList:=[op(exprList),[tempList[i],"1"]]:
          else exprList:=[op(exprList),[tempList[i],"0"]]:
          end if:
      else
          for n from xPos to length(tempStr) do
             if tempStr[n]="*" or tempStr[n]=")" then n:=n-1:
             break:  end if:
          end do:
          exprList:=[op(exprList),[tempList[i],substring
                  (tempStr,xPos+2..n)]]:
      end if:
end do:
for i from 1 to nops(exprList) do
      minPos:=i:
      for j from i+1 to nops(exprList) do
          if length(exprList[minPos][2])>length(exprList[j]
               [2]) then minPos:=j:
          elif length(exprList[minPos][2])=length(exprList
               [j][2]) and exprList[minPos][2]>exprList[j][2]
               then minPos:=j:
          end if:
      end do:
      temp:=exprList[minPos]:
      exprList[minPos]:=exprList[i]:
      exprList[i]:=temp:
      resList:=[op(resList)+temp[1]]:
end do:
      return op(resList):
```

```
end proc:
```

国际通用计算机代数系统Maple为进行科学计算提供了一个较大的平台, 但无论它的功能多么强大, 不可能也无法实现所有的科学计算, 即使能够完成的工作也可能存在着一些漏洞. 这就需要我们在研究过程中能够利用相关数学基础设计相应的算法弥补这些不足. 这或许是在通用计算机代数系统下进行数学机械化需要注意的一个重要问题.

第 4 章　非线性积分 – 微分方程求解及其机械化

文献 [184] 研究了如下形式的非线性 Volterra-Fredholm 积分方程:

$$y(x) = f(x) + \lambda_1 \int_a^x k_1(x,t)[y(t)]^p dt + \lambda_2 \int_a^b k_2(x,t)y(t)dt,$$

而文献 [185] 考虑了如下形式的高阶线性 Volterra-Fredholm 积分 — 微分方程:

$$\sum_{j=0}^{m} P_j(x)y^{(j)}(x) = f(x) + \lambda_1 \int_a^x k_1(x,t)y(t)dt + \lambda_2 \int_a^b k_2(x,t)y(t)dt,$$

本章, 我们将考虑如下所示的高阶非线性 Volterra-Fredholm 积分 – 微分方程:

$$\sum_{j=0}^{m} P_j(x)y^{(j)}(x) = f(x) + \lambda_1 \int_a^x k_1(x,t)[y(t)]^p dt + \lambda_2 \int_a^b k_2(x,t)[y(t)]^q dt, \qquad (4.1)$$

其中 $P_j(x)(j=0,1,\cdots,m)$, $f(x)$, $k_1(x,t)$, $k_2(x,t)$ 是在区间 $a \leqslant x,\ t \leqslant b$ n 致 $(n \geqslant m)$ 阶可导函数, $a, b, \lambda_1, \lambda_2$ 是常数, p, q 是正整数. 方程 (4.1) 的解可表述为如下形式:

$$y(x) = \sum_{j=0}^{N} \frac{1}{n!} y^{(n)}(\xi)(x-\xi)^n, \quad a \leqslant x, \xi \leqslant b, \quad N \geqslant m. \qquad (4.2)$$

这是一个在 $x = \xi$ 处的 N 阶 Taylor 多项式函数, 这里 $y^{(n)}(\xi)$ $(n = 0, 1, \cdots, N)$ 是待定系数.

4.1　基本算法

参考文献 [184, 185] 中的算法设计, 本节, 我们将建立求解高阶非线性 Volterra-Fredholm 积分 – 微分方程 (4.1) 的算法.

首先, 我们将 (4.1) 改写为:

$$D(x) = f(x) + \lambda_1 V(x) + \lambda_2 F(x) \quad 或 \quad D(x) = I(x) \qquad (4.3)$$

其中

$$D(x) = \sum_{j=0}^{m} P_j(x) y^{(j)}(x),$$
$$V(x) = \int_a^x k_1(x,t)[y(t)]^p dt, \tag{4.4}$$
$$F(x) = \int_a^b k_2(x,t)[y(t)]^q dt,$$

$$I(x) = f(x) + \lambda_1 V(x) + \lambda_2 F(x). \tag{4.5}$$

这里, $D(x)$ 称为微分部分, $I(x)$ 称为积分部分.

要得到形式如 (4.2) 的解, 我们必须将 (4.1) 即 (4.3) 关于 x 求导 n 次, 得:

$$D^{(n)}(x) = I^{(n)}(x), \tag{4.6}$$

为了讨论方便, 记:

$$D^{(0)}(x) = D(x), \quad I^{(0)}(x) = I(x).$$

下面, 分析表达式 $D^{(n)}(x)$ 和 $I^{(n)}(x)$.

易知:

$$D^{(n)}(x) = [P_0(x)y(x)]^{(n)} + [P_0(x)y'(x)]^{(n)} + \cdots + \left[P_m(x)y^{(m)}(x)\right]^{(n)}, \quad (n = 0, 1, \cdots, N,\ N \geqslant m). \tag{4.7}$$

应用 Leibnitz 规则 (处理函数乘积的导数的法则), 将 (4.7) 简化并代入 $x = \xi$, 可得:

$$D^{(n)}(\xi) = \sum_{j=0}^{m} \sum_{k=0}^{n} \binom{n}{k} P_j^{(n-k)}(\xi) y^{(k+j)}(\xi),\ (n = 0, 1, \cdots, N). \tag{4.8}$$

系统 (4.8) 可改写为矩阵形式:

$$\mathbf{D} = \mathbf{W}\mathbf{Y}, \tag{4.9}$$

其中,

$$\mathbf{Y} = \left[y^{(0)}(\xi) \quad y^{(1)}(\xi) \quad \cdots \quad y^{(N)}(\xi) \right]^T \tag{4.10}$$

是要确定的. 而

$$\mathbf{W} = [W_{nk}], \quad n, k = 0, 1, \cdots, N, \tag{4.11}$$

矩阵 $\mathbf{W}$ 中的元素由下式决定:

$$W_{nk} = \sum_{s=0}^{m} \binom{n}{k-m+s} P_{m-s}^{(n-k+m-s)}(\xi). \tag{4.12}$$

若 $r<0$, 则

$$P_0^{(r)}(\xi)=P_1^{(r)}(\xi)=\cdots=P_m^{(r)}(\xi)=0,$$

若 $j<0$ 且 $j>i$, $\begin{pmatrix} i \\ j \end{pmatrix}=0$, 这里 i, j, r 是整数. 在这种情况下, 方程 (4.12) 对于 $n<k-m$ $(n=0,1,\cdots,N-m-1;\ k=m+1,m+2,\cdots,N)$, 均有:

$$W_{nk}=0.$$

另一方面, 对于积分部分 $I(x)$, 易知当 $n\geqslant 1$ 时,

$$I^{(n)}(x)=f^{(n)}(x)+\lambda_1 V^{(n)}(x)+\lambda_2 F^{(n)}(x), \tag{4.13}$$

其中,

$$\begin{aligned} V^{(n)}(x)&=\tfrac{d^n}{dx^n}\left(\int_a^x k_1(x,t)[y(t)]^p dt\right), \\ F^{(n)}(x)&=\int_a^b \tfrac{\partial^n k_2(x,t)}{\partial x^n}[y(t)]^q dt. \end{aligned} \tag{4.14}$$

记:

$$Y_1(t)=[y(t)]^p,\quad Y_2(t)=[y(t)]^q. \tag{4.15}$$

将式 (4.15) 代入 (4.14), 可得:

$$\begin{aligned} V^{(n)}(x) &= \frac{d^n}{dx^n}\left(\int_a^x k_1(x,t)Y_1(t)dt\right) \\ &= \sum_{i=0}^{n-1}[k_1(x,t)Y_1(x)]^{(n-i-1)}+\int_a^x \frac{\partial^n k_1(x,t)}{\partial x^n}Y_1(t)dt \\ &= \sum_{k=0}^{n-1}\sum_{i=0}^{n-k-1}\begin{pmatrix} n-i-1 \\ k \end{pmatrix} h_i^{(n-k-i-1)}(x)Y_1^{(k)}(x) \\ &+ \int_a^x \frac{\partial^n k_1(x,t)}{\partial x^n}Y_1(t)dt \end{aligned} \tag{4.16}$$

其中,

$$h_i(x)=\left.\frac{\partial^i k_1(x,t)}{\partial x^i}\right|_{t=x}.$$

而

$$F^{(n)}(x)=\int_a^b \frac{\partial^n k_2(x,t)}{\partial x^n}Y_2(t)dt. \tag{4.17}$$

利用 Taylor 展开定理, 可以得到:

$$Y_i(t)=\sum_{k=0}^{\infty}\frac{1}{k!}Y_i^{(k)}(\xi)(t-\xi)^k \quad (i=1,2). \tag{4.18}$$

应用 Leibnitz 规则并将 (4.18) 分别代入 (4.16) 和 (4.17) 可得:

$$\begin{aligned} V^{(n)}(\xi) &= \sum_{k=0}^{n-1}\sum_{i=0}^{n-k-1}\begin{pmatrix} n-i-1 \\ k \end{pmatrix} h_i^{(n-k-i-1)}(\xi)Y_1^{(k)}(\xi) \\ &+ \int_a^{\xi} \left.\frac{\partial^n k_1(x,t)}{\partial x^n}\right|_{x=\xi}\left[\sum_{k=0}^{\infty}\frac{1}{k!}Y_1^{(k)}(\xi)(t-\xi)^k\right]dt, \end{aligned} \tag{4.19}$$

$$F^{(n)}(\xi) = \int_a^b \left.\frac{\partial^n k_2(x,t)}{\partial x^n}\right|_{x=\xi}\left[\sum_{k=0}^{\infty}\frac{1}{k!}Y_2^{(k)}(\xi)(t-\xi)^k\right]dt. \tag{4.20}$$

由此可得:

$$\begin{aligned} I^{(n)}(\xi) &= f^{(n)}(\xi) + \lambda_1 V^{(n)}(\xi) + \lambda_2 F^{(n)}(\xi) \\ &= f^{(n)}(\xi) + \lambda_1\sum_{k=0}^{n-1}\sum_{i=0}^{n-k-1}\begin{pmatrix} n-i-1 \\ k \end{pmatrix} h_i^{(n-k-i-1)}(\xi)Y_1^{(k)}(\xi) \\ &+ \lambda_1\int_a^{\xi} \left.\frac{\partial^n k_1(x,t)}{\partial x^n}\right|_{x=\xi}\left[\sum_{k=0}^{\infty}\frac{1}{k!}Y_1^{(k)}(\xi)(t-\xi)^k\right]dt \\ &+ \lambda_2\int_a^{b} \left.\frac{\partial^n k_2(x,t)}{\partial x^n}\right|_{x=\xi}\left[\sum_{k=0}^{\infty}\frac{1}{k!}Y_2^{(k)}(\xi)(t-\xi)^k\right]dt \\ &= f^{(n)}(\xi) + \lambda_1\left\{\sum_{k=0}^{n-1} H_{nk}Y_1^{(k)}(\xi) + \sum_{k=0}^{\infty} T_{nk}Y_1^{(k)}(\xi)\right\} \\ &+ \lambda_2\sum_{k=0}^{\infty} K_{nk}Y_2^{(k)}(\xi), \end{aligned} \tag{4.21}$$

在实际计算过程中, 我们可使用下式作为 (4.21) 的近似式:

$$\begin{aligned} I^{(n)}(\xi) &= f^{(n)}(\xi) + \lambda_1\left\{\sum_{k=0}^{n-1}(H_{nk}+T_{nk})\,Y_1^{(k)}(\xi) + \sum_{k=n}^{N} T_{nk}Y_1^{(k)}(\xi)\right\} \\ &+ \lambda_2\sum_{k=0}^{N} K_{nk}Y_2^{(k)}(\xi). \end{aligned} \tag{4.22}$$

当 $n=1,2,\cdots;\ k=0,1,\cdots,n-1\ (n>k)$ 时, 对于 H_{nk} 有:

$$H_{nk} = \sum_{i=0}^{n-k-1}\begin{pmatrix} n-i-1 \\ k \end{pmatrix} h_i^{(n-k-i-1)}(c), \tag{4.23}$$

当 $n \leqslant k$ 时,

$$H_{nk} = 0,$$

当 $n,k=0,1,2,\cdots$ 时, 对于 T_{nk} 和 K_{nk} 分别有:

$$T_{nk}=\frac{1}{k!}\int_a^{\xi}\left.\frac{\partial^n k_1(x,t)}{\partial x^n}\right|_{x=\xi}(t-\xi)^k dt, \tag{4.24}$$

$$K_{nk}=\frac{1}{k!}\int_a^{b}\left.\frac{\partial^n k_2(x,t)}{\partial x^n}\right|_{x=\xi}(t-\xi)^k dt. \tag{4.25}$$

而

$$Y_i^{(n)}(\xi)=\sum_{t_1+t_2+\cdots+t_s=n}\begin{pmatrix} n \\ t_1t_2\cdots t_s\end{pmatrix}y^{(t_1)}(\xi)y^{(t_2)}(\xi)\cdots y^{(t_s)}(\xi)\quad(i=1,2), \tag{4.26}$$

其中, $t_j\,(j=1,2,\cdots,s)$ 为非负整数且:

$$\begin{pmatrix} n \\ t_1t_2\cdots t_s\end{pmatrix}=\frac{n!}{t_1!\ t_2!\ \cdots t_s!}.$$

事实上, 利用公式 (4.26), 我们可将 $Y_i(\xi)$ $(i=1,2)$ 转化为要确定的系数

$$y^{(0)}(\xi),\, y^{(1)}(\xi),\cdots,y^{(N)}(\xi)$$

的形式.

据此, 如果 $n,k=0,1,2,\cdots,N$, 可得:

$$\begin{aligned}
&I^{(0)}(\xi)=f^{(0)}(\xi)+\lambda_1\textstyle\sum_{k=0}^{N}T_{0k}Y_1^{(k)}(\xi)+\lambda_2\sum_{k=0}^{N}K_{0k}Y_2^{(k)}(\xi),\\
&I^{(n)}(\xi)=f^{(n)}(\xi)+\lambda_1\left\{\textstyle\sum_{k=0}^{n-1}\left(H_{nk}+T_{nk}\right)Y_1^{(k)}(\xi)+\sum_{k=n}^{N}T_{nk}Y_1^{(k)}(\xi)\right\}\\
&\qquad+\lambda_2\textstyle\sum_{k=0}^{N}K_{nk}Y_2^{(k)}(\xi)\quad(n=1,2,\cdots;\ k=0,1,\cdots,N).
\end{aligned} \tag{4.27}$$

系统 (4.27) 可写成如下矩阵形式:

$$\mathbf{I}=\mathbf{F}+\lambda_1\mathbf{TY_1^*}+\lambda_2\mathbf{KY_2^*}. \tag{4.28}$$

考虑到 (4.9) 与 (4.28), 我们可得:

$$\mathbf{WY}=\mathbf{F}+\lambda_1\mathbf{TY_1^*}+\lambda_2\mathbf{KY_2^*}. \tag{4.29}$$

其中,

$$\mathbf{W}=\begin{bmatrix} W_{00} & \cdots & W_{0m} & 0 & 0 & \cdots & 0 \\ W_{10} & \cdots & W_{1m} & W_{1,m+1} & 0 & \cdots & 0 \\ W_{20} & \cdots & W_{2m} & W_{2,m+1} & W_{2,m+2} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ W_{N-m-1,0} & \cdots & W_{N-m-1,m} & W_{N-m-1,m+1} & W_{N-m-1,m+2} & \cdots & 0 \\ W_{N-m,0} & \cdots & W_{N-m,m} & W_{N-m,m+1} & W_{N-m,m+2} & \cdots & W_{N-m,N} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ W_{N0} & \cdots & W_{Nm} & W_{N,m+1} & W_{N,m+2} & \cdots & W_{NN} \end{bmatrix}$$

$$\mathbf{F}=\left[f^{(0)}(c) \quad f^{(1)}(c) \quad \cdots f^{(N)}(c)\right]^T,$$

$$\mathbf{K}=\begin{bmatrix} K_{00} & K_{01} & K_{02} & \cdots & K_{0N} \\ K_{10} & K_{11} & K_{12} & \cdots & K_{1N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ K_{N0} & K_{N1} & K_{N2} & \cdots & K_{NN} \end{bmatrix},$$

$$\mathbf{T}=\begin{bmatrix} T_{00} & T_{01} & T_{02} & \cdots & T_{0N} \\ H_{10}+T_{10} & T_{11} & T_{12} & \cdots & T_{1N} \\ H_{20}+T_{20} & H_{21}+T_{21} & T_{12} & \cdots & T_{1N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ H_{N0}+T_{N0} & H_{N1}+T_{N1} & H_{N2}+T_{N2} & \cdots & T_{NN} \end{bmatrix}, \tag{4.30}$$

$$\mathbf{Y}_1^*=\left[Y_1^{(0)}(c) \quad Y_1^{(1)}(c) \quad \cdots Y_1^{(N)}(c)\right]^T,$$

$$\mathbf{Y}_2^*=\left[Y_2^{(0)}(c) \quad Y_2^{(1)}(c) \quad \cdots Y_2^{(N)}(c)\right]^T.$$

而记:

$$\mathbf{Y}=\left[y^{(0)}(c) \quad y^{(1)}(c) \quad \cdots y^{(N)}(c)\right]^T \overset{\mathrm{d}}{=} [y_0 \quad y_1 \quad \cdots \quad y_N]^T, \tag{4.31}$$

就是我们要求的 (4.2) 中的 Taylor 多项式函数的系数.

将 (4.30) 代入 (4.29), 并将其转换为关于 $y_0, y_1, \cdots, y_N$ 的代数方程, 求解代数方程组即可确实 Taylor 系数 $y^{(n)}(\xi)$ $(n=0,1,\cdots,N)$, 于是我们可得方程 (4.1) 的

解为:

$$y(x)=\sum_{n=0}^{N}\frac{1}{n!}y^{(n)}(\xi)(x-\xi)^n. \tag{4.32}$$

注 1: 如果选择 $\xi=a$, 则系统 (4.28) 变为:

$$\begin{aligned}&I^{(0)}(a)=f^{(0)}(a)+\lambda_2\textstyle\sum_{k=0}^{N}K_{0k}Y_2^{(k)}(a),\\&I^{(n)}(a)=f^{(n)}(a)+\lambda_1\textstyle\sum_{k=0}^{n-1}H_{nk}Y_1^{(k)}(\xi)+\lambda_2\textstyle\sum_{k=0}^{N}K_{nk}Y_2^{(k)}(\xi).\end{aligned} \tag{4.33}$$

考虑到方程 (4.30), 则方程 (4.1) 的解为:

$$y(x)=\sum_{n=0}^{N}\frac{1}{n!}y^{(n)}(a)(x-a)^n. \tag{4.34}$$

注 2: 若 $\lambda_1=0$, 则系统 (4.1) 即为 Fredholm 型积分 — 微分方程; 如果 $\lambda_2=0$, 则系统 (4.1) 即为 Volterra 型积分 — 微分方程.

注 3: 在系统 (4.1) 中, 若 $\lambda_1=\lambda_2=0$, 则 (4.1) 即为常微分方程.

4.2 机械化算法

从上节的推导过程特别是公式 (4.29) 的计算可知, 利用本文所建立的算法求解高阶非线性 Volterra-Fredholm 积分 — 微分方程时计算量很大. 因此, 在`Maple`下建立基于上节结果的机械化求解 (4.1) 的算法显得十分必要.

公式 (4.3)— (4.33) 可方便地用于在计算机代数系统`Maple`下建立求解高阶非线性 Volterra-Fredholm 积分 — 微分方程 (4.1) 的机械化算法, 整个求解 (4.1) 的 Taylor 多项式解的过程都可在`Maple`下进行算法设计和程序设计. 而运用机械化算法求解方程 (4.1) 时, 我们所要做的所有事情就是输入方程的信息, 机械化算法将会自动给出方程 (4.1) 的 Taylor 多项式解, 并输出主要求解步骤. 主程序`voltfredproc`列示如下:

```
voltfredproc:=proc(expr, N, c)
local lexpr, rexpr, m, p, i, j, f, temp, pint, k, y_ba,
    a, b, lambd, Y, TT, W, K, F, y_star, T, A, B, Eqn,
    equn,CONST;
CONST:=[exp,ln,log,abs,sin,cos,tan,cot,sec,scs];
Y:=matrix(N+1,1,[seq(y[i],i=0..N)]);
lexpr:=lhs(expr); lambd:=[ ]; rexpr:=rhs(expr);
```

```
k:=[ ]; y_star:=[ ]; y_ba:=[ ];
m:=degdiff(lexpr,y(x),x); p:=[ ];
for i from 0 to m do
    p:=[op(p),coefdiff(lexpr,y(x),x,i)];
end do;
temp:=[op(rexpr)]; f:=[ ]; pint:=[ ];
for i from 1 to nops(temp) do
    if not has(temp[i],int) then
      f:=[op(f),temp[i]];
    else
      pint:=[op(pint),temp[i]];
    end if;
od;
f:=convert(f,`+`);
if nops(pint)=0 then
    lambd:=[op(lambd),0];
    a:=0; b:=0; k:=[0,0]; y_ba:=[0];
    y_star:=[op(y_star),matrix(N+1,1,Yorother(unapply
        (y_ba[1],t)(x),N))]
else
    for i from 1 to nops(pint) do
      temp:=parmeint(pint[i]);
      if nops(pint)=1 and type(temp[2,2],numeric) then
      y_ba:=[op(y_ba),0]; k:=[op(k),0];
      lambd:=[op(lambd),0];
      y_star:=[op(y_star),0];
      i:=i+1;
    fi;
    y_ba:=[op(y_ba),temp[1,2]]; k:=[op(k),temp[1,1]];
    if i=1 then a:=temp[2,1]; fi;
    if i=2 then
      b:=temp[2,2]; a:=temp[2,1];
    else
      b:=0;
    fi;
```

```
    lambd:=[op(lambd),temp[3]];
    y_star:=[op(y_star),matrix(N+1,1,Yorother(unapply
        (y_ba[i],t)(x),N))];
    od;
    if nops(k)=1 then k:=[op(k),0]; fi;
fi;
for i from 1 to nops(CONST) do
    if has(f,CONST[i]) then
      f:=convert(taylor(f,x=0,2*N),polynom);
    fi;
    for j from 1 to nops(k) do
      if has(k[j],CONST[i]) then
        k[j]:=convert(taylor(k[j],x=0,2*N),polynom);
      fi;
    od;
    for j from 1 to nops(p) do
      if has(p[j],CONST[i]) then
        p[j]:=convert(taylor(p[j],x=0,2*N),polynom);
      fi;
    od;
od;
W:=Wmatrix(N, p, m, c); T:=Tmatrix(N, k[1], c, a);
F:=Fvector(f, N); K:=Kmatrix(N, k[2], a, b, c);
A:=multiply(W, Y); TT:=[T, K];
if y_ba[1]=0 then
    B:=evalm(F);
else
    B:=evalm(F+sum(lambd[Bi]*(TT[Bi]&*y_star[Bi]),Bi=1..
         nops(lambd)));
fi;
Eqn:={ };
for i from 1 to N+1 do
    Eqn:={op(Eqn),A[i,1]=B[i,1]};
od;
lprint(`The matrix form of the equation is as follows:`);
```

```
print(WY=F+lambd[1]*TY[1]+lambd[2]*TY[2]);
lprint('where');
print('W'=W,'T'=T);
print('F'=Matrix(F),'K'=K);
lprint('Then we can get the following equations:');
print(Vector(op(Eqn)));
lprint('The solution of this equations is as follows:');
equn:=[solve(Eqn,seq(y[i],i=0..N))];
if nops(equn)=1 then equn:=op(equn); else equn:=equn[1]; fi;
print('Y'=equn);
lprint('Then we can obtain the Taylor polynomial solution
     of the equation:');
print('y(x)'=subs(op(equn),sum(1/n!*y[n]*(x-c)^n,
      'n'=0..N)));
end:
```

在主程序voltfred中, 参数集的含义如下:

expr: 要求解的方程;
N: Taylor 多项式解的阶数;
ξ: 要求解的方程关于变量的 Taylor 级数展开点.

例如, 对于下述 Volterra-Fredholm 积分 — 微分方程:

$$y'' + xy' - 3y = e^x - 2\int_{-1}^{x} \sin xy^3(t)dt + \int_{-1}^{1} \sin xe^{-t}y^2(t)dt$$

在voltfred下求解此方程所做的全部事情就是在Maple中输入关于此方程的信息:

```
expr:=diff(y(x),x$2)+x*diff(y(x),x)-3*y(x)=exp(x)-2*int
      (sin(x)*y^3(t),t=-1..x)+int(sin(x)*exp(-t)*y^2(t),
      t=-1..1);
voltfredproc(expr, 5, 0);
```

则机械化算法voltfred即可输出该方程在 $x=0$ 处的 5 阶 Taylor 多项式解.

另外, voltfred 中包含了一些子程序, 诸如Wmatrix, Tmatrix, Fvector, Kmatrix 等, 它们的作用是计算 (4.30) 中对应的矩阵 $\mathbf{W}, \mathbf{T}, \mathbf{F}, \mathbf{K}$. 作为例子, 在此列示Tmatrix 的代码如下:

```
Tmatrix:=proc(N, k1, c, a)
local T, H, T1, n, k, i, temp;
T:=Matrix(N+1,N+1);
H:=Matrix(N+1,N+1);
T1:=Matrix(N+1,N+1);
for n from 0 to N do
    for k from 0 to N do
      if n<=k then H[n+1, k+1]:=0
      else
        for i from 0 to n-k-1 do
          if i=0 then
            temp:=unapply(k1, t);
          else
            temp:=unapply(binomial(n-i-1,k)*diff(k1,x$i),t);
          fi;
          if n-k-i-1>0 then
            temp:=unapply(diff(temp(x),x$(n-k-i-1)),x);
          else
            temp:=unapply(temp(x),x);
          fi;
          H[n+1,k+1]:=H[n+1,k+1]+binomial(n-i-1,k)*temp(c);
        od;
      fi;
      if n=0 then
        temp:=unapply(k1, x);
      else
        temp:=unapply(diff(k1, x$n), x);
      fi;
      T1[n+1,k+1]:=1/k!*int(temp(c)*(t-c)^k,t=a..c);
      if k>=n then
        T[n+1,k+1]:=T1[n+1,k+1];
      else
        T[n+1,k+1]:=T1[n+1,k+1]+H[n+1,k+1];
      fi;
    od;
```

```
od;
    return(T);
end:
```

其余子程序的算法设计与此类似, 不再赘述.

4.3 应用算例

本节, 我们通过下述例子说明voltfred的机械化求解过程及其高效性.

例 4.1: 我们首先考虑如下 Volterra-Fredholm 积分 — 微分方程:

$$xy''-xy'+2y=\frac{1}{12}x^4-\frac{1}{6}x^3-\frac{1}{2}x^2-\frac{13}{6}x+\frac{17}{12}+\int_0^x(x-t)y(t)dt+\int_0^1(x+t)y(t)dt. \tag{4.35}$$

在Maple下, 我们只需要输入以下关于方程 (4.35) 的信息描述的命令:

```
expr1:=x*diff(y(x),x$2)-x*diff(y(x),x)+2*y(x)= 1/12*x^4
      -1/6*x^3-x^2/2-13/6*x+17/12+ int((x-t)*y(t),t=0..x)
      +int((x+t)*y(t),t=0..1);
voltfredproc(expr1,5,0);
```

通过机械化算法voltfredproc即可得到方程 (4.35) 的求解过程及结果:

```
The matrix form of the equation is as follows:
```

$$WY=F+\lambda_1 TY_1+\lambda_2 TY_2.$$

```
where
```

$$W=\begin{bmatrix}2&0&0&0&0&0\\0&1&1&0&0&0\\0&0&0&2&0&0\\0&0&0&-1&3&0\\0&0&0&0&-2&4\\0&0&0&0&0&-3\end{bmatrix},\quad T=\begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\end{bmatrix},$$

$$F = \begin{bmatrix} \frac{17}{12} \\ -\frac{13}{6} \\ -1 \\ -1 \\ 2 \\ 0 \end{bmatrix}, \quad K = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{8} & \frac{1}{30} & \frac{1}{144} & \frac{1}{840} \\ 1 & \frac{1}{2} & \frac{1}{6} & \frac{1}{24} & \frac{1}{120} & \frac{1}{720} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

`Then we can get the following equations:`

$$2\,y_0 = \tfrac{17}{12} + \tfrac{1}{2}\,y_0 + \tfrac{1}{3}\,y_1 + \tfrac{1}{8}\,y_2 + \tfrac{1}{30}\,y_3 + \tfrac{1}{144}\,y_4 + \tfrac{1}{840}\,y_5$$

$$y_1 + y_2 = -\tfrac{13}{6} + y_0 + \tfrac{1}{2}\,y_1 + \tfrac{1}{6}\,y_2 + \tfrac{1}{24}\,y_3 + \tfrac{1}{120}\,y_4 + \tfrac{1}{720}\,y_5$$

$$2\,y_3 = -1 + y_0$$

$$-y_3 + 3\,y_4 = -1 + y_1$$

$$-2\,y_4 + 4\,y_5 = 2 + y_2$$

$$-3\,y_5 = y_3$$

`The solution of this equations is as follows:`

$$Y = \{y_0 = 1, y_1 = 1, y_3 = 0, y_4 = 0, y_5 = 0, y_2 = -2\}$$

`Then we can obtain the Taylor polynomial solution of the equation:`

$$y(x) = 1 + x - x^2$$

易验证 $y(x) = 1 + x - x^2$ 是方程 (4.35) 的解析解.

通过这个例子可以看到, 为了可读性, 在主程序`voltfredproc`中, 我们运用命令`lprint`和`print` 设计了可读性求解过程. 这样, 通过`voltfredproc`求解高阶非线性 Volterra-Fredholm 积分 — 微分方程的过程就像我们用纸和笔求解过程一样. 这种可读性输出结果对于计算机辅助求解方程是有益的.

应用`Maple`下的计时命令`time`可获知求解此方程约需要 9.8 秒. 由此, 可见该算法的高效性.

例 4.2: 考虑方程:

$$x^2y^{(4)} - y'' + y' = \frac{1}{2} + \frac{1}{2}x^2 - \frac{1}{30}\,x^6 - \frac{1}{6}\,x^4 + \frac{8}{3}\,x + \int_0^x (x-t)y(t)^2dt - 2\int_0^1 (x+t)y(t)dt \tag{4.36}$$

在Maple中, 输入下述命令:

```
expr2:=x^2*diff(y(x),x$4)-diff(y(x),x$2)+y(x)=1/2+1/2*x^2
       -1/30*x^6-1/6*x^4+8/3*x +int(((x-t)*y(t)^2),t=0..x)
       -2*int((x+t)*y(t),t=0..1);
voltfredproc(expr2,3,0);
```

即可得到如下结果:

The matrix form of the equation is as follows:

$$WY = F + \lambda_1 TY_1 + \lambda_2 TY_2$$

where

$$W = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} \frac{1}{2} \\ \frac{8}{3} \\ 1 \\ 0 \end{bmatrix}, \quad K = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{8} & \frac{1}{30} \\ 1 & \frac{1}{2} & \frac{1}{6} & \frac{1}{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Then we can get the following equations:

$$y_0 - y_2 = \tfrac{1}{2} - y_0 - \tfrac{2}{3} y_1 - \tfrac{1}{4} y_2 - \tfrac{1}{15} y_3$$

$$y_1 - y_3 = \tfrac{8}{3} - 2 y_0 - y_1 - \tfrac{1}{3} y_2 - \tfrac{1}{12} y_3$$

$$y_2 = 1 + {y_0}^2$$

$$y_3 = 2 y_0 y_1$$

The solution of this equations is as follows:

$$Y = \{y_3 = 0, y_1 = 0, y_0 = 1, y_2 = 2\}$$

Then we can obtain the Taylor polynomial solution of the equation:

$$y(x) = 1 + x^2$$

显然, $y(x)=1+x^2$ 是方程 (4.36) 的解析解.

另一方面, Taylor 多项式的阶数 N 如果选择太大, 则计算复杂性高; 若 N 太小, 则计算的误差可能会增大. 例如, 在例 4.2 求解过程中, 如果我们取 $N=2$ 时, 算法voltfredproc 给出的 (4.36) 的结果为:

$$y(x)=\frac{17}{31}+\frac{546}{961}x+\frac{625}{961}x^2.$$

由此可见, 我们必须在Maple下通过实验的手段选择恰当的 N 以保证解的精确度. 在此, 我们将通过下面一个简单的例子来说明数学实验及相应的结果.

例 4.3: 考虑

$$y'=1-\int_0^x y(t)dt. \tag{4.37}$$

输入命令:

```
expr3:=diff(y(x),x)=1-int(y(t),t=0..x);
voltfredproc(expr3,5,0);
```

这里, 我们取 $N=5$. 结果为:

```
The matrix form of the equation is as follows:
```

$$WY=F+\lambda_1 TY_1+\lambda_2 TY_2$$

```
where
```

$$W=\begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix},\quad T=\begin{bmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\end{bmatrix}$$

$$F = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Then we can get the following equations:

$$y_1 = 1$$

$$y_2 = -y_0$$

$$y_3 = -y_1$$

$$y_4 = -y_2$$

$$y_5 = -y_3$$

$$0 = -y_4$$

The solution of this equations is as follows:

$$Y = \{y_1 = 1, y_2 = 0, y_4 = 0, y_3 = -1, y_5 = 1, y_0 = 0\}$$

Then we can obtain the Taylor polynomial solution of the equation:

$$y(x) = x - \frac{1}{6}x^3 + \frac{1}{120}x^5$$

若取 $N = 7$, 则可得方程 (4.37) 的解为:

$$y(x) = x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7$$

若 $N = 11$:

$$y(x) = x - \frac{1}{6}x^3 + \frac{1}{120}x^5 - \frac{1}{5040}x^7 + \frac{1}{362880}x^9 - \frac{1}{39916800}x^{11}$$

若 $N = 15$:

$$y(x) = x - \tfrac{1}{6}x^3 + \tfrac{1}{120}x^5 - \tfrac{1}{5040}x^7 + \tfrac{1}{362880}x^9 - \tfrac{1}{39916800}x^{11} + \tfrac{1}{6227020800}x^{13} - \tfrac{1}{1307674368000}x^{15}$$

事实上, (4.37) 的解析解为

$$y(x) = \sin x,$$

而 $\sin x$ 在 $x = 0$ 处的 Taylor 展开式为:

$$\sin x = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}.$$

显然, 解的精度依赖于 N 的大小.

例 4.4: 考虑方程:

$$y''' + y(x) = -\frac{1}{5}x^5 + \frac{2}{3}x^3 + \frac{5}{6}x^2 - \frac{113}{105}x - 1 + \int_0^x y^2(t)dt + \int_0^1 \left(x^2t + xt^2\right) y^2(t)dt \tag{4.38}$$

在Maple中输入关于方程 (4.38) 的信息:

```
expr4:=diff(y(x),x$3)+y(x)=5*x^2/6-8*x/105-x-1/5*x^5
        +2/3*x^3-1+int(y(t)^2,t=0..x)+int((x^2*t+x*t^2)
        *(y(t))^2,t=0..1);
voltfredproc(expr4, 4, 0);
```

求解结果为:

```
The matrix form of the equation is as follows:
```

$$WY = F + \lambda_1 TY_1 + \lambda_2 TY_2$$

```
where
```

$$W = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} -1 \\ -\frac{113}{105} \\ \frac{5}{3} \\ 4 \\ 0 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{10} & \frac{1}{36} & \frac{1}{168} \\ 1 & \frac{2}{3} & \frac{1}{4} & \frac{1}{15} & \frac{1}{72} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Then we can get the following equations:

$$y_0 + y_3 = -1$$

$$y_1 + y_4 = -\tfrac{113}{105} + \tfrac{4}{3}\,y_0{}^2 + \tfrac{1}{2}\,y_0y_1 + \tfrac{1}{5}\,y_1{}^2 + \tfrac{1}{5}\,y_0y_2 + \tfrac{1}{6}\,y_1y_2 + \tfrac{1}{18}\,y_0y_3 + \tfrac{1}{28}\,y_2{}^2 + \tfrac{1}{21}\,y_1y_3 + \tfrac{1}{84}\,y_0y_4$$

$$y_2 = \tfrac{5}{3} + \tfrac{10}{3}\,y_0y_1 + y_0{}^2 + \tfrac{1}{2}\,y_1{}^2 + \tfrac{1}{2}\,y_0y_2 + \tfrac{2}{5}\,y_1y_2 + \tfrac{2}{15}\,y_0y_3 + \tfrac{1}{12}\,y_2{}^2 + \tfrac{1}{9}\,y_1y_3 + \tfrac{1}{36}\,y_0y_4$$

$$y_3 = 4 + 2\,y_1{}^2 + 2\,y_0y_2$$

$$y_4 = 6\,y_1y_2 + 2\,y_0y_3$$

The solution of this equations is as follows:

$$Y = \{y_2 = 2, y_0 = -1, y_3 = 0, y_4 = 0, y_1 = 0\}$$

Then we can obtain the Taylor polynomial solution of the equation:

$$y(x) = -1 + x^2$$

显然, 这是方程 (4.38) 的解析解. 但是, 在该例中, 如果我们取 $N = 3$ 则得不到任何结果, 而当 $N = 5$ 时, 尽管我们可以得到解析解 $y(x) = -1 + x^2$, 但计算耗费的时间稍多. 从这个意义上讲, 对例 4.4 来说, $N = 4$ 是合适的阶数选择.

例 4.5: 考虑积分方程:

$$y(x) = -\frac{15}{56}x^8 + \frac{13}{14}x^7 - \frac{11}{10}x^6 + \frac{9}{20}x^5 + x^2 - x + \int_0^x (x+t)y^3(t)dt \tag{4.39}$$

在Maple下输入:

```
expr5:=y(x)=-15/56*x^8+13/14*x^7-11/10*x^6+9/20*x^5+x^2
```

```
            -x+int((x+t)*(y(t))^3,t=0..x);
voltfredproc(expr5, 5, 0);
```

可得机械化求解过程为:

```
The matrix form of the equation is as follows:
```

$$WY = F + \lambda_1 TY_1 + \lambda_2 TY_2$$

```
where
```

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 0 \\ 54 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
Then we can get the following equations:
```

$$y_0 = 0$$

$$y_3 = 15\, {y_0}^2 y_1$$

$$y_4 = 21\, y_0 \left(2\, {y_1}^2 + y_0 y_2\right)$$

$$y_5 = 54 + 54\, {y_1}^3 + 162\, y_0 y_1 y_2 + 27\, {y_0}^2 y_3$$

$$y_1 = -1$$

$$y_2 = 2 + 3\, {y_0}^3$$

```
The solution of this equations is as follows:
```

$$Y = \{y_0 = 0, y_3 = 0, y_5 = 0, y_4 = 0, y_1 = -1, y_2 = 2\}$$

```
Then we can obtain the Taylor polynomial solution of the
        equation:
```

$$y(x) = -x + x^2$$

此为 (4.39) 的解析解.

例 4.6: 求解方程:

$$y(x) = e^x - \frac{1}{3}e^{3x} + \frac{1}{3} + \int_0^x y^3(t)dt \tag{4.40}$$

在Maple中输入:

```
expr6:=y(x)=exp(x)-1/3*exp(3*x)+1/3+int((y(t)^3),t=0..x);
voltfredproc(expr6,7,0);
```

即可得如下求解结果:

```
The matrix form of the equation is as follows:
```

$$WY = F + \lambda_1 TY_1 + \lambda_2 TY_2$$

```
where
```

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$F=\begin{bmatrix}1\\0\\-2\\-8\\-26\\-80\\-242\\-728\end{bmatrix},\quad K=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Then we can get the following equations:

$$y_0=1$$

$$y_1={y_0}^3$$

$$y_3=-8+3\,y_0\,(2\,{y_1}^2+y_0y_2)$$

$$y_4=-26+6\,{y_1}^3+18\,y_0y_1y_2+3\,{y_0}^2y_3$$

$$y_5=-80+36\,{y_1}^2y_2+18\,y_0{y_2}^2+24\,y_0y_1y_3+3\,{y_0}^2y_4$$

$$y_6=-242+90\,y_1{y_2}^2+60\,{y_1}^2y_3+60\,y_0y_2y_3+30\,y_0y_1y_4+3\,{y_0}^2y_5$$

$$y_7=-728+90\,{y_2}^3+360\,y_1y_2y_3+90\,{y_1}^2y_4+60\,y_0{y_3}^2+90\,y_0y_2y_4+36\,y_0y_1y_5 \\ +3\,{y_0}^2y_6$$

$$y_2=-2+3\,{y_0}^2y_1$$

The solution of this equations is as follows:

$$Y=\{y_0=1,y_1=1,y_7=1,y_6=1,y_3=1,y_2=1,y_4=1,y_5=1\}$$

Then we can obtain the Taylor polynomial solution of the equation:

$$y(x)=1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\frac{1}{720}x^6+\frac{1}{5040}x^7$$

易证 (4.40) 的解析解为:

$$y(x)=e^x=1+x+\frac{1}{2}x^2+\frac{1}{6}x^3+\frac{1}{24}x^4+\frac{1}{120}x^5+\frac{1}{720}x^6+\frac{1}{5040}x^7+O(x^8),$$

其中 $O(x^8)$ 为余项 — 刻划了解析解与 Taylor 多项式解之间的误差.

4.4　高阶常微分方程求解

在方程 (4.1) 中, 如果参数 λ_1, λ_2 皆为 0, 此时, 微分 — 积分方程即退化为常微分方程. 显然, 机械化算法voltfredproc可用于求解高阶非线性常微分方程.

下面, 我们通过两个算例展示机械化算法voltfredproc在求解 ODE 时的应用.

例 4.7: 考虑常微分方程:

$$\frac{d^4}{dx^4}y(x) + (1-x)\,y(x) = 24 + x^2(1-x)^3 \tag{4.41}$$

在Maple下输入命令:

```
expr7:=diff(y(x),x$4)+(1-x)*y(x)=24+x^2*(1-x)^3;
voltfredproc(expr7,6,0);
```

即可得求解过程:

```
The matrix form of the equations is as follows:
```

$$WY = F + \lambda_1 TY_1 + \lambda_2 TY_2$$

where

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 \\ 0 & 0 & -3 & 1 & 0 \\ 0 & 0 & 0 & -4 & 1 \end{bmatrix}, \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} 24 \\ 0 \\ 2 \\ -18 \\ 72 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
Then we can get the following equations:
```

$$y_0 + y_4 = 24$$

$$-3\,y_2 + y_3 = -18$$

$$-4\,y_3 + y_4 = 72$$

$$-y_0 + y_1 = 0$$

$$-2\,y_1 + y_2 = 2$$

```
The solution of this equations is as follows:
```

$$Y = \{y_4 = 24, y_3 = -12, y_2 = 2, y_1 = 0, y_0 = 0\}$$

```
Then we can obtain the Taylor polynomial solution of the
        equation:
```

$$y(x) = x^2 - 2x^3 + x^4,$$

即为方程 (4.41) 的解析解.

虽然, 在Maple中包含了求解常微分方程的函数dsolve, 但却无法求解方程 (4.41). 这也说明了我们建立的算法voltfredproc用于求解常微分方程时的功能较dsolve强大.

例 4.8: 求解方程:

$$x^2 \frac{d^5}{dx^5} y(x) - \frac{d^2}{dx^2} y(x) + y(x) = \frac{1}{2} + \frac{1}{2} x^2 - \frac{1}{6} x^4 \tag{4.42}$$

输入命令:

```
expr8:=x^2*diff(y(x),x$5)-diff(y(x),x$2)+y(x)
    =1/2+1/2*x^2-1/6*x^4;
mechodeproc(expr6,5,0);
```

可得求解结果为:

```
The matrix equation of the ODE is as follows:
```

$$WY = F$$

where

$$W = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 2x \\ 0 & 0 & 1 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad F = \begin{bmatrix} \frac{1}{2} \\ 0 \\ 1 \\ 0 \\ -4 \\ 0 \end{bmatrix}$$

Then we can get the following equations:

$$y_0 - y_2 = \tfrac{1}{2}$$

$$y_1 - y_3 + 2\,xy_5 = 0$$

$$y_2 - y_4 + 2\,y_5 = 1$$

$$y_3 - y_5 = 0$$

$$y_4 = -4$$

$$y_5 = 0$$

The solution of this equations is as follows:

$$Y = \left\{ y_0 = -\frac{5}{2},\ y_1 = 0,\ y_2 = -3,\ y_3 = 0,\ y_4 = -4,\ y_5 = 0 \right\}$$

Then we can obtain the Taylor polynomial solution of the equation:

$$y(x) = -\frac{5}{2} - \frac{3}{2}x^2 - \frac{1}{6}x^4$$

即为 (4.42) 的解析解.

4.5 本章小结

本章研究了高阶非线性 Volterra-Fredholm 积分 — 微分方程的 Taylor 多项式解的算法, 在此基础上, 建立了基于Maple平台的机械化算法voltfredproc, 利用该算法可以获得此类积分 — 微分方程的 Taylor 多项式解, 有时该解即为解析解. 应用实例表示该算法是求解高阶非线性 Volterra-Fredholm 积分 — 微分方程的高效算法之一. 这将为积分 — 微分方程求解提供参考.

另一方面, 研究结果表明, 数学机械化是积分 — 微分方程求解的重要方法之一. 计算机高效快捷的特点能够帮助人们完成复杂的计算. 数学机械化的基础是算法设计和程序设计, 其中, 算法设计是程序设计的前提. 在程序设计中, 为了追求高效快捷, 我们必须建立尽可能简单的算法, 但是, 为了保证高精度, 我们又不得不使用更为复杂的算法. 这是一对无法调和的矛盾.

第 5 章　结束语

近三十年来, 计算机技术的迅猛发展为数学研究提供了坚实的支撑, 特别是吴文俊院士所倡导的 “数学机械化” 的研究思想和方法为众多数学问题提供了可行的解决方案. 本书关于 Fredholm 积分方程 (组), Volterra 积分方程 (组) 以及高阶 Volterra-Fredholm 积分 — 微分方程求解及其机械化算法的研究当属一次有益的尝试, 也是数学机械化思想在积分方程求解方面的具体实践.

数学机械化的基础是算法设计和程序设计, 其中, 算法设计是程序设计的前提. 在程序设计中, 为了追求高效快捷, 我们必须建立尽可能简单的算法, 但是, 为了保证高精度, 我们又不得不使用更为复杂的算法.

研究结果表明, 数学机械化是积分方程求解的有效方法之一, 计算机高效快捷的特点能够帮助人们完成复杂的计算. 这将为其他类型方程求解提供参考.

级数法是方程求解中一种常用的方法, 运用这一方法通常可以获得方程的近似解, 而近似程度取决于级数展开的阶数. 一个值得研究的问题是如何利用有限项的级数解获取级数的通项, 并进而无穷求和从而求得方程的精确解是. 本书对此进行了探讨, 并成功运用于 Volterra 积分方程机械化求解中. 而当我们将这一做法用于其他问题的解决时, 发现因为重重困难而无法得到我们期望的结果, 甚至于我们无法证明对于任意函数是否存在某种形式的级数表达. 这将是我们后续研究的一个问题.

另外一个值得研究的问题是如何将本书已有的成果推广至其他类型的积分方程的机械化求解中, 诸如 Cauchy 核奇异积分方程

$$A(t)\varphi(t)+\frac{1}{\pi i}\int_L\frac{k(x,\tau)}{\tau-t}\varphi(t)=\eta(t),\ t\in L,$$

Wiener-Hopf 型方程

$$\varphi(x)-\int_0^\infty k(x-y)\varphi(y)dy=\psi(x),\ 0\leqslant x<\infty$$

以及随机积分方程、非线性奇异积分方程组等. 相对于本书所研究的 Fredholm 积分方程 (组), Volterra 积分方程 (组) 以及高阶 Volterra-Fredholm 积分 — 微分方程求解而言, 这些积分方程的求解更为复杂, 涉及到泛函分析和拓朴学等复杂数学分支, 同时, 这类方程求解涉及边值问题. 这也将是我们后续研究的一个重点.

另一方面, 我们在研究中发现, 国际通用计算机代数系统Maple为进行科学计算提供了一个功能强大的平台. 但是, Maple毕竟是一个软件, 它不可能也无法实现所有的科学计算, 即使能够完成的工作也可能存在着一些漏洞. 这就需要我们在研究过程中能够利用相关数学基础设计相应的算法弥补这些不足. 事实上, 这也是我们在通用计算机代数系统下进行数学机械化研究时需要注意的一个重要问题.

附录A 符号积分研究及应用

本章内容是作者最早进入数学机械化领域的研究成果 [142, 143], 虽与本书主体略有不符, 但所涉及内容属于 “高等数学” 最基本的东西, 读者对其相当熟悉, 因而, 通过本章研究结果, 读者能更清楚地理解符号计算及数学机械化过程中算法建立的基本思想和方法, 故作为附录列于本书中.

A.1 问题的提出

计算机科学以及计算机代数系统的迅速发展对数学研究的观念及研究方法都产生了深刻的影响, 可以说这种影响是一场革命性的影响. 而微积分的创立, 更是被恩格斯誉为 “人类精神的最高胜利”. 18 世纪后, 微积分进一步深入发展, 这种发展与广泛的应用紧密交织在一起, 刺激和推动了许多数学分支的产生和发展 [66]. 积分的计算特别是符号积分计算是微积分及计算机数学领域中的一个难点. 在微分方程求解, 有关定理自动推证等问题中, 随处可见符号积分计算.

diff, int函数分别是计算机代数系统`Maple`中的求导函数和积分函数. 尽管如此, 在其下关于符号积分依然存在缺陷, 例如:

>Diff(f(x)*g(x),x)=diff(f(x)*g(x),x);

$$\frac{\partial}{\partial x}\left(f\left(x\right)g\left(x\right)\right)=\left(\frac{\partial}{\partial x}f\left(x\right)\right)g\left(x\right)+f\left(x\right)\frac{\partial}{\partial x}g\left(x\right)$$

为了便于阅读, 本章中 > 后的直体英文表示在`Maple`下的输入, 而斜体的表示`Maple`下的输出结果. 在`Maple 7` 下, 微分符号统一用 ∂, 这一点虽然看起来不是很习惯, 但在程序设计中却因为少一个记号而节约空间. 但`Maple 8` 以后的版本已经将一元函数的导数形式改为 $\frac{dy}{dx}$, 而偏微分依然记为 ∂.

但对上式再积分, `Maple`却只返回原表达式:

>int(rhs(%),x);

$$\int\left(\frac{\partial}{\partial x}f\left(x\right)\right)g\left(x\right)+f\left(x\right)\left(\frac{\partial}{\partial x}g\left(x\right)\right)dx$$

其中,% 表示上一次执行结果, 本文中均指上一行运算结果. **rhs** (right hand side) 表示等号右端表达式, 相应地, 等号左端表达式用命令`lhs` (left hand side).

另一方面, 应用更为广泛的计算机代数系统Matlab的符号计算核心是Maple, 因此同样存在上述问题. 基于此, 本章综合运用计算机代数理论和方法, 提出新的符号积分算法, 以期解决计算机代数系统下符号积分问题.

A.2 符号积分算法设计

A.2.1 原函数为乘积表达式

定理 1: 函数组 $\{f_1(x), f_2(x), ..., f_n(x)\}$ 中任取 $m(m \leqslant n)$ 个函数, 不妨设这 m 个函数为 $L = \{f_1(x), f_2(x), ..., f_m(x)\}$, 对于 $\forall f_i(x) \in L$ 的 p_i 次方的乘积表达式:

$$F(x) = \prod_{i=1}^{m} f_i(x)^{p_i} \ (p_i \in R),$$

的导函数为:

$$F'(x) = F_1'(x) + F_2'(x) + ... + F_i'(x) + ... + F_m'(x),$$

则 $F'(x)$ 中至少存在一个操作数 $F_i'(x)\,(i \in \{1, 2, ...m\})$, 将其导数降低 1 阶后与 $F(x)$ 至多相差一个常数因子.

证明: 由函数乘积的求导公式 $(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$ 即可得证.

为了更清楚地理解定理 1 的内容, 下面给出两个例子.

例 1:

$$F_1(x) = f_1^3(x)f_2^9(x)x^3$$

对函数 $F_1(x)$ 求导得:

$$F_1'(x) = 3f_1^2(x)f_1'(x)f_2^9(x)x^3 + 9f_1^3(x)f_2^8(x)f_2'(x)x^3 + 3f_1^3(x)f_2^9(x)x^2$$

其操作数分别为:

$$3f_1^2(x)f_1'(x)f_2^9(x)x^3\ ,\ 9f_1^3(x)f_2^8(x)f_2'(x)x^3\ ,\ 3f_1^3(x)f_2^9(x)x^2$$

上式中前两个操作数去掉导数符号后分别为 $3f_1^3(x)f_2^9(x)x^3, 9f_1^3(x)f_2^9(x)x^3$, 与 $F_1(x)$ 只相差常数分别为 3, 9.

例 2:

$$F_2(x) = f_1'^3(x)f_2^9(x)$$

对函数 $F_2(x)$ 求导得:

$$F_2'(x) = 3\left(\frac{\partial}{\partial x}f_1(x)\right)^2 f_2^9(x)\left(\frac{\partial^2}{\partial x^2}f_1(x)\right) + 9\left(\frac{\partial}{\partial x}f_1(x)\right)^3 f_2^8(x)\left(\frac{\partial}{\partial x}f_2(x)\right)$$

其操作数分别为:

$$3\left(\frac{\partial}{\partial x}f_1(x)\right)^2 f_2^9(x)\left(\frac{\partial^2}{\partial x^2}f_1(x)\right),\ 9\left(\frac{\partial}{\partial x}f_1(x)\right)^3 f_2^8(x)\left(\frac{\partial}{\partial x}f_2(x)\right)$$

上式每项导数降低一阶后分别为: $3\left(\frac{\partial}{\partial x}f_1(x)\right)^3 f_2^9(x), 9\left(\frac{\partial}{\partial x}f_1(x)\right)^3 f_2^9(x)$, 与 F_1 相比只差常数为 3, 9.

另外, 对于有理式或有理函数Maple通常会自动化简, 即消去分子, 分母中的公约数或公因式. 但对于更复杂的含有无理数或无理式的多项式却不能进行这种化简. 如下例:

例 3: 求下面两函数的比值:

$$c := \sqrt{3}\left(\frac{\partial}{\partial x}f(x)\right) + g(x)$$

$$d := 3\left(\frac{\partial}{\partial x}f(x)\right) + \sqrt{3}g(x)$$

```
> c/d;
```

$$\frac{\sqrt{3}\left(\frac{\partial}{\partial x}f(x)\right) + g(x)}{3\left(\frac{\partial}{\partial x}f(x)\right) + \sqrt{3}g(x)}$$

利用Maple下的强行化简命令simplify或者normal也得到同样的结果, 均未得到我们想象的 $\frac{\sqrt{3}}{3}$. 为了解决该问题, 我们给出下述定理. 定理中所谓的函数多项式是指 + 项至少为 2 项的函数表达式, 而后文中的函数单项式指无有 + 项的函数表达式.

定理 2: 任给两个同类函数多项式 $F_1(x), F_2(x)$, 设 L_1, L_2 是 $F_1(x), F_2(x)$ 分别展开后所有操作数组成的列表, 若 $F_1(x) = cF_2(x)$, 则对 $\forall F_{1i}(x) \in L_1, \exists F_{2j}(x) \in L_2$, 使得 $F_{1i}(x) = cF_{2j}(x)$ (其中 c 为常数).

证明: 利用 *op* 获取两个函数多项式的操作数列表 L_1, L_2, 再将 L_1, L_2 中的操作数两两比较即可得证.

由定理 2 可知, 要判定两个函数多项式 $F_1(x), F_2(x)$ 是否相差一个常数因子, 可以通过寻找函数多项式 $F_1(x)$ 在 $F_2(x)$ 中的对应项获取. 例如 $F_{11}(x)$ 为 $F_1(x)$ 中

的任一项, $F_2(x)$ 中与之对应的项为 $F_{21}(x)$, 当两者相差一个常数因子, 即存在常数 $c = \frac{F_{11}(x)}{F_{21}(x)}$ 时, 只需判定函数 $F_1(x), cF_2(x)$ 是否相等即可 (在Maple中判断两个表达式是否相等很容易).

基于上述结果, 我们在计算机代数系统Maple下建立了原函数为乘积表达式的符号积分算法myint1, 程序伪码如下:

```
> myint1:=proc(expr,var::name)
    if 表达式 expr 可用 int 积分 or expr 是单项式 then
        return int(expr,var);
    end if:
    对 expr 的每一个含有导数的单项式将导数运算降低 1 阶后保存到列表 LR 中;
    for i from 1 to nops(LR) do
        if diff(LR[i],var) 整除 expr, 商为常数 c then
            return LR[i]*c;
        end if:
    end do:
    return int(express,var);
end proc:
```

调用程序myint1, 可以解决一类符号积分问题:

例 4:

(1) > diff(f(x)*g(x),x);

$$\left(\frac{\partial}{\partial x}f(x)\right)g(x)+f(x)\left(\frac{\partial}{\partial x}g(x)\right)$$

> myint1(%,x);

$$f(x)g(x)$$

(2) > diff(f(x)^3*g(x)^(sqrt(3))*diff(h(x),x),x);

$$3\,(f(x))^2\,g(x)\left(\frac{\partial}{\partial x}h(x)\right)\frac{\partial}{\partial x}f(x)+(f(x))^3\left(\frac{d}{dx}g(x)\right)\frac{\partial}{\partial x}h(x)+(f(x))^3\,g(x)\,\frac{\partial^2}{\partial x^2}h(x)$$

> myint1(%,x);

$$(f(x))^3\,(g(x))^{\sqrt{3}}\,\frac{\partial}{\partial x}h(x)$$

(3) > diff(f(x)^2*g(x)^(-1)*h(x)^(sqrt(2)),x);

$$2\,\frac{f(x)\,(h(x))^{\sqrt{2}}\,\frac{d}{dx}f(x)}{g(x)}-\frac{(f(x))^2\,(h(x))^{\sqrt{2}}\,\frac{d}{dx}g(x)}{(g(x))^2}+\frac{(f(x))^2\,(h(x))^{\sqrt{2}}\,\sqrt{2}\frac{d}{dx}h(x)}{g(x)\,h(x)}$$

> myint1(%,x);

$$\frac{(f(x))^2\,(h(x))^{\sqrt{2}}}{g(x)}$$

(4) > myint1(diff(f(x)*g(x)*diff(h(x)^2,x),x);

$$2\left(\frac{\partial}{\partial x}f(x)\right)g(x)\,h(x)\,\frac{\partial}{\partial x}h(x)+2\,f(x)\left(\frac{\partial}{\partial x}g(x)\right)h(x)\,\frac{\partial}{\partial x}h(x)$$

$$+2\,f(x)\,g(x)\left(\frac{\partial}{\partial x}h(x)\right)^2+2\,f(x)\,g(x)\,h(x)\,\frac{\partial^2}{\partial x^2}h(x)$$

> myint1(,x);

$$2\,f(x)\,g(x)\,h(x)\,\frac{\partial}{\partial x}h(x)$$

(5) > x*diff(f(x),x$2)+diff(f(x),x);　　# $ 表示高阶导数, $2 表示求 2 阶导数

$$x\frac{\partial^2}{\partial x^2}f(x)+\frac{\partial}{\partial x}f(x)$$

>myint1(%,x);

$$x\frac{\partial}{\partial x}f(x)$$

但是, 下述例子表明该程序依然存在缺陷:

例 5:

> diff(f(x)*g(x),x)+1;

$$\frac{\partial}{\partial x}f(x)\,g(x)+f(x)\,\frac{\partial}{\partial x}g(x)+1$$

```
> myint1(%,x);
```

$$\int \frac{\partial}{\partial x} f(x)\, g(x) + f(x)\, \frac{\partial}{\partial x} g(x) + 1 \, dx;$$

由此可见, 算法myint1虽然能够成功解决原函数为乘积表达式的积分问题, 但不能解决类似上式的符号积分问题, 为了解决该问题, 我们给出原函数为任意函数多项式的算法.

A.2.2　原函数为任意函数多项式

任意函数多项式

$$F(x) = \sum_{i=1}^{n} F_i(x) = \sum_{i=1}^{n}\prod_{j=1}^{m} f_{ij}^{p_{ij}}(x) \quad (p_{ij} \in R)$$

求导后可得:

$$F'(x) = \sum_{i=1}^{n} F_i'(x) = \sum_{i=1}^{n}\sum_{k=1}^{n_i}\prod_{j=1, j\neq k}^{n_i} p_{ik} f_{ij}^{p_{ij}}(x) f_{ik}^{(p_{ik}-1)}(x) f_{ik}' \quad (n_i \in Z; p_{ij}, p_{ik} \in R)$$

若 $F_i(x)$ 是 $F_i'(x)$ 的一个原函数, 显然对 $F_i'(x)$ 的积分问题可以类似上节中对 $F'(x)$ 中的项 $F_i'(x)$ 求积分方法解决. 但是, $F'(x)$ 中的项 $F_i'(x)$ 与 $F_j'(x)$ 可能相互交叉, 从而使问题复杂化. 解决该问题的关键是找到 $F_i'(x)$ 的所有操作数, 在此, 我们采用贪婪法和回溯法设立了两个列表 *intop*, *outop* 分别存储可积分项和不可积分项的所有操作数. 为了更清楚地表达我们的算法设计思路, 下面给出一个简单算例:

例 6:

$$K := xf''(x) + 2f'(x)$$

对 K 积分的正确方法是将 K 分为两项 $(xf''(x) + f'(x)) + f'(x)$ 后再分项积分, 但 myint1 的算法未能考虑到这一点. 因此, 必须对 myint1 算法进行修改. 修改的思路如下: 先将 K 的第一个操作数降阶为 $xf'(x)$, 再将求导结果 $xf''(x) + f'(x)$, 与 K 比较 (应用自行设计子程序完成), 并返回三项结果:$[true, 1, f'(x)]$, 第一项说明两个函数是否有共同部分, 第二项为两函数的商, 第三项为余式.

在上述思考的基础上, 我们设计了用于单项式函数符号积分的子程序singleint:

```
> singleint:=proc(expr,var::name)
    if 表达式子 expr 可用 int 积分 then
    return [int(expr,var),0];
```

```
    end if:
    对 expr 的每一个含有导数的单项式将导数运算降低 1 阶后保存到列表 LR 中;
    for i from 1 to nops(LR) do
        if diff(LR[i],var) 相除 expr 的商为常数 c then
            return [c*diff(LR[i],var),diff(LR[i],var) 除 expr 余式];
        end if;
    end do:
    return [int(expr,var),0];
end proc;
```

由此可见, 该函数的主要功能是判断 expr 能否与一个单项式的导函数同类 (即组成项只是系数不同).

调用`singleint`对例 6 中的 K 积分, 结果如下:

>singleint(K,x);

$$[x\frac{\partial}{\partial x}f(x), \frac{\partial}{\partial x}f(x)]$$

多项式函数可看作是单项式函数经过 $+,-,*,/$ 以及复合而得. 在`singleint`的基础上, 结合前述算法, 我们设计了解决符号积分问题的程序`myint`, 程序伪码如下:

```
> myint:=proc(expr,var::name)
    if 表达式 expr 可用 int 积分 then
        return int(expr,var);
    end if;
    if expr 是某单项式的导函数 then
        return singleint(expr,var)[1];
    end if;
    用 intop 存放 expr 所有已积分项, 用 outop 存放 expr 所有未积分项;
    while nops(outop)<> 0 do
        i:=1;                              # 表示在 outop 中取 i 个元素
        while i<=nops(outop) do
            调用 num 得到 outop 所有长度为 i 的子列表;
        if singleint(每一子列表组成的函数多项式, var)[2]<>0 then
            在 outop 中删除该子列表并在 intop 中加入该子列表;
            break;
        else
            i:=i+1;
```

```
        end if;
        end do;
        if outop 所有子列表都可积分 then
            return singleint(expr,var)[1];
        if outop 的所有子列表都不可积分 then
            break;
        end if;
    end do;
    return convert(intop, +)+int(convert(outop, +),var);
end proc:
```

应用该程序可以解决符号积分问题. 下面通过实例说明myint的用法及功能. 首先给出前面提到的两个问题的myint积分结果:

例 7:

(1) > diff(f(x)*g(x),x)+1;

$$\frac{\partial}{\partial x}f(x)\,g(x)+f(x)\,\frac{\partial}{\partial x}g(x)+1$$

>myint(%,x);

$$x+f(x)*g(x)$$

(2) > x*diff(f(x),x$2)+2*diff(f(x),x);

$$x\frac{d^2}{dx^2}f(x)+2\,\frac{d}{dx}f(x)$$

>myint(%,x);

$$x\frac{\partial}{\partial x}f(x)+f(x)$$

对于较复杂的函数的积分, myint也是能够快速完成的:

例 8:

(1) > diff(g(x),x$2)*diff(f(x),x)^2+2*diff(g(x),x)*diff(f(x),x)*diff(f(x),x$2);

$$\left(\frac{\partial^2}{\partial x^2}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)^2+2\left(\frac{\partial}{\partial x}f(x)\right)\left(\frac{\partial}{\partial x}g(x)\right)\frac{\partial^2}{\partial x^2}f(x)$$

> myint(%,x);

$$\left(\frac{\partial}{\partial x}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)^2$$

(2) > diff(f(x)/(w(x)+g(x)),x);

$$\frac{\frac{\partial}{\partial x}f(x)}{w(x)+g(x)}-\frac{f(x)\left(\frac{\partial}{\partial x}w(x)+\frac{\partial}{\partial x}g(x)\right)}{(w(x)+g(x))^2}$$

对上式求积分:

> myint(%,x);

$$\frac{f(x)}{w(x)+g(x)}$$

(3) >diff((f[1](x)+f[2](x))/(f[3](x)+f[4](x)),x);

$$\frac{\frac{\partial}{\partial x}f_1(x)+\frac{\partial}{\partial x}f_2(x)}{f_3(x)+f_4(x)}-\frac{(f_1(x)+f_2(x))\left(\frac{\partial}{\partial x}f_3(x)+\frac{\partial}{\partial x}f_4(x)\right)}{(f_3(x)+f_4(x))^2}$$

对上式求积分, 得:

> myint(%,x);

$$\frac{f_1(x)}{f_3(x)+f_4(x)}+\frac{f_2(x)}{f_3(x)+f_4(x)}$$

对于含高阶导数的符号积分, `myint`也可以胜任.

例 9:

(1) > diff(f(x)*g(x), x$3);

$$\left(\frac{\partial^3}{\partial x^3}g(x)\right)f(x)+3\left(\frac{\partial^2}{\partial x^2}g(x)\right)\frac{\partial}{\partial x}f(x)+3\left(\frac{d^2}{dx^2}f(x)\right)\frac{\partial}{\partial x}g(x)+g(x)\frac{\partial^3}{\partial x^3}f(x)$$

对其积分:

> myint(%, x);

$$\left(\frac{\partial^2}{dx^2}g(x)\right)f(x)+2\left(\frac{\partial}{\partial x}g(x)\right)\frac{\partial}{\partial x}f(x)+g(x)\frac{\partial^2}{dx^2}f(x)$$

对上式再积分:

> myint(%, x);

$$\left(\frac{\partial}{\partial x}g(x)\right)f(x)+g(x)\frac{\partial}{\partial x}f(x)$$

再积分:

> myint(%,x);

$$g(x)\,f(x)$$

(2) > diff(f(x)/g(x),x$2);

$$\frac{\frac{\partial^2}{\partial x^2}f(x)}{g(x)}-2\,\frac{\left(\frac{\partial}{\partial x}g(x)\right)\frac{\partial}{\partial x}f(x)}{(g(x))^2}+2\,\frac{f(x)\left(\frac{\partial}{\partial x}g(x)\right)^2}{(g(x))^3}-\frac{\left(\frac{\partial^2}{\partial x^2}g(x)\right)f(x)}{(g(x))^2}$$

积分后, 可得:

> myint(%,x);

$$\frac{\frac{\partial}{\partial x}f(x)}{g(x)}-\frac{\left(\frac{\partial}{\partial x}g(x)\right)f(x)}{(g(x))^2}$$

再对上式积分, 得:

> myint(%,x);

$$\frac{f(x)}{g(x)}$$

(3) > diff(diff(g(x),x)*diff(f(x),x)^2,x$2);

$$\left(\frac{\partial^3}{\partial x^3}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)^2+4\left(\frac{\partial^2}{\partial x^2}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)\frac{\partial^2}{\partial x^2}f(x)$$
$$+2\left(\frac{\partial}{\partial x}g(x)\right)\left(\frac{\partial^2}{\partial x^2}f(x)\right)^2+2\left(\frac{\partial}{\partial x}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)\frac{\partial^3}{\partial x^3}f(x)$$

再积分, 得:

> myint(%,x);

$$\left(\frac{\partial^2}{\partial x^2}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)^2+2\left(\frac{\partial}{\partial x}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)\frac{\partial^2}{\partial x^2}f(x)$$

再对上式积分, 得:

> myint(%,x);

$$\left(\frac{\partial}{\partial x}g(x)\right)\left(\frac{\partial}{\partial x}f(x)\right)^2$$

对于复合函数的情形, `myint`也能完成:

例 10:

(1) > diff(f(x)^g(x)*k(x),x);

$$(f(x))^{g(x)}\left(\left(\frac{\partial}{\partial x}g(x)\right)\ln(f(x))+\frac{g(x)\frac{\partial}{\partial x}f(x)}{f(x)}\right)k(x)+(f(x))^{g(x)}\frac{\partial}{\partial x}k(x)$$

对上式积分:

```
> myint(%,x);
```

$$(f(x))^{g(x)}k(x)$$

再看下面更复杂的复合函数:

```
(2) > diff(f(x)^g(k(x))*m(x),x);
```

$$(f(x))^{g(k(x))}\left(\mathrm{D}(g)(k(x))\left(\frac{\partial}{\partial x}k(x)\right)\ln(f(x))+\frac{g(k(x))\frac{\partial}{\partial x}f(x)}{f(x)}\right)m(x)$$
$$+(f(x))^{g(k(x))}\frac{\partial}{\partial x}m(x)$$

(其中, D是`Maple`下的微分算子).

对其积分, 可得:

```
> myint(%,x);
```

$$f(x)^{g(k(x))}m(x)$$

当给出具体函数解析式时可看作是符号函数的特殊情形, 因此总可以用`myint`积分. 虽然具体函数积分在某些情况下`int`也可以完成, 但对于`int`不能积分的函数, `myint`也可以积分.

例 11:

```
(1) > diff(x^sin(x)+f(x),x);
```

$$x^{\sin(x)}\left(\cos(x)\ln(x)+\frac{\sin(x)}{x}\right)+\frac{d}{dx}f(x)$$

```
> int(%,x);
```

$$x^{\sin(x)}+f(x)$$

```
> myint(%,x);
```

$$x^{\sin(x)}+f(x)$$

```
(2) > G(x):=diff(x^exp(x)*f(x),x);
```

$$G(x)=x^{e^x}\left(e^x\ln(x)+\frac{e^x}{x}\right)f(x)+x^{e^x}\frac{\partial}{\partial x}f(x)$$

> int(G(x),x);

$$\int x^{e^x}\left(e^x\ln(x)+\frac{e^x}{x}\right)f(x)+x^{e^x}\frac{\partial}{\partial x}f(x)\,dx$$

> myint(G(x),x);

$$x^{e^x}f(x)$$

而当遇到无法积分的符号表达式时, myint会将可积分部分表达式积分, 而将不能积分部分的表达式返回:

例 12:

>diff(f(x)*g(x),x)+h(x)*k(x);

$$\left(\frac{\partial}{\partial x}g(x)\right)f(x)+g(x)\frac{\partial}{\partial x}f(x)+h(x)k(x)$$

>myint(%,x);

$$f(x)g(x)+\int h(x)k(x)\,dx$$

A.3 符号积分在微分中值定理自动推证中的应用

微分学中的 Rolle 定理, Lagrange 中值定理, Cauchy 中值定理, 介值定理, 零点存在定理和各种引人入胜的推广及应用, 不仅在微积分学中占有极为重要的地位, 也引起了很多数学家的注意, 象 Polya, Poincare, Schwarz, Landau, Huiwitz 等世界一流数学家在这一方面也有工作 [83].

下面, 我们将给出微分中值定理自动推证.

A.3.1 微分中值定理自动推证的基本思想

在讨论问题前, 先写下熟知的几个定理. 以下均假设 $f(x)$, $g(x)$ 满足:1° 在 $[a,b]$ 连续; 2° 在 (a,b) 可导.

Rolle **定理**: 若 $f(a)=f(b)$, 则存在 $\xi\in(a,b)$, 使

$$f'(\xi)=0$$

Lagrange **定理**: 存在 $\xi\in(a,b)$, 使

$$f'(\xi)=\frac{f(b)-f(a)}{b-a}$$

Cauchy **定理**: 若 $g'(x) \neq 0$, 则存在 $\xi \in (a,b)$, 使

$$\frac{f'(\xi)}{g'(\xi)} = \frac{f(b)-f(a)}{g(b)-g(a)}$$

关于 Lagrange 定理与 Cauchy 定理的证明, 分别引入辅助函数

$$F(x) = f(x) - \frac{f(b)-f(a)}{b-a}x \tag{A.1}$$

$$F(x) = f(x) - \frac{f(b)-f(a)}{g(b)-g(a)}g(x) \tag{A.2}$$

并称辅助函数 (A.1) 是考查曲线 $f(x)$ 的切线而作出的. 而对辅助函数 (A.2) 未讲做法, 有一些教材称 (A.2) 是类比 (A.1) 得到的.

实际上, 对于函数 (A.1), (A.2) 可作如下考虑. Rolle 定理换句话说就是: 在 $f(a) = f(b)$ 的条件下, 证明 $f'(x)$ 在 (a,b) 内有零点. 所以当考虑 Lagrange 定理时, 即证 $f'(x) - \frac{f(b)-f(a)}{b-a}$ 在 (a,b) 内有零点. 而由 Rolle 定理知道, 这只需要其原函数 $f(x) - \frac{f(b)-f(a)}{b-a}x$ 在 $[a,b]$ 区间端点 a,b 处的函数值相等. 此乃积分辅助函数法之本质 [83].

对于 Cauchy 中值定理来说, 其原表达式直接积分不可能. 这时, 需要将表达式化为等价形式:

$$f'(\xi) = \frac{f(b)-f(a)}{g(b)-g(a)}g'(\xi)$$

再积分即可得到辅助函数 (A.2).

A.3.2 微分中值定理自动推证 Maple 程序

根据上述辅助函数积分法思想即可写出如下微分中值定理自动推证程序, 其中的`myint`即为前述符号积分函数.

```
> myproc:=proc(function, name::symbol)
    local L, R, intL, intR, F, Ldenom, Lnumer, Rdenom, Rnumer;
    L:=lhs(function);               # 取出 function 的左边部分并赋值给 L
    R:=rhs(function);               # 取出 function 的右边部分并赋值给 R
    Ldenom:=denom(L);               # 取出 L 的分母部分
    Lnumer:=numer(L);               # 取出 L 的分子部分
    Rdenom:=denom(R);
    Rnumer:=numer(R);
    if not has(myint(L, name), int) and not has(myint(R, name), int) then
        intL:=myint(L, name);
```

```
        intR:=myint(R, name);
        F:=intL−intR;
        if has(L, int) or has(R, int) then
            F:=myint(Lnumer−R*Ldenom, name);
        end if;
    else
        intL:=myint(Lnumer*Rdenom, name);
        intR:=myint(Rnumer*Ldenom, name);
        if has(intL, int) or has(intR, int) then
            F:=myint(Lnumer*Rdenom−Rnumer*Ldenom, name);
        else
            F:=intL−intR;
        end if;
    end if;
    lprint('辅助函数为:');
    print('F(x)'=F);
    F:=unapply(F,x);
    if is(simplify(F(a)−F(b))=0) then
        print ('F(a)−F(b)'=simplify(expand(F(a))−expand(F(b))));
        lprint('由于 F(a)=F(b), F(x) 满足 Rolle 定理条件, 于是有:');
        print('F'(x)'=diff(F(name), name));
        lprint('据此可得:');
        print(fuction);
        lprint('命题得证.');
    else
        print('对不起, 此题无法用此系统证明 ');
    fi;
end proc:
```

A.3.3 微分中值定理自动推证实现

在Maple下, 运行我们设计的微分中值定理自动推证程序myproc, 即可获得推证过程. 运行程序时, 只需将 ξ 改写为 x 即可.

例 13: 若 $f(x),g(x)$ 在 $[a,b]$ 连续, (a,b) 可导, 则存在 $\xi\in(a,b)$, 使

$$f(b)-f(a)=f'(\xi)(b-a) \tag{A.3}$$

> myproc(f(b)−f(a)=diff(f(x),x)*(b−a), x);

此为Maple下的输入语句, 接下来的是Maple的自动推证结果. 下同.

辅助函数为:

$$F(x) = x(f(b) - f(a)) - (b-a)f(x)$$

$$F(a) - F(b) = 0$$

由于 $F(a) = F(b)$, $F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x) = f(b) - f(a) - \left(\frac{d}{dx}f(x)\right)(b-a)$$

据此可得:

$$f(b) - f(a) = \left(\frac{d}{dx}f(\xi)\right)(b-a)$$

命题得证.

例 14: 若 $f(x), g(x)$ 在 $[a,b]$ 连续, (a,b) 可导, 且 $g'(x) \neq 0$, 则存在 $\xi \in (a,b)$, 使

$$\frac{f'(\xi)}{g'(\xi)} = \frac{f(b)-f(a)}{g(b)-g(a)} \tag{A.4}$$

> myproc(diff(f(x), x)/diff(g(x), x)=(f(b)-f(a))/(g(b)-g(a)), x);

辅助函数为:

$$F(x) = f(x) - \frac{(f(b)-f(a))g(x)}{g(b)-g(a)}$$

$$F(a) - F(b) = 0$$

由于 $F(a) = F(b), F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x) = \frac{d}{dx}f(x) - \frac{(f(b)-f(a))\frac{d}{dx}g(x)}{g(b)-g(a)}$$

据此可得:

$$\frac{\frac{d}{dx}f(\xi)}{\frac{d}{dx}g(\xi)} = \frac{f(b)-f(a)}{g(b)-g(a)}$$

命题得证.

例 15: 若 $f(x)$ 在 $[a,b]$ 连续, 在 (a,b) 可导, 则存在 $\xi \in (a,b)$, 使

$$(b^2-a^2)f'(\xi) = 2\xi(f(b)-f(a)) \tag{A.5}$$

```
> myproc((b^2-a^2)*diff(f(x), x)=2*x*(f(b)-f(a)), x);
```

辅助函数为:

$$F(x) = \left(b^2-a^2\right) f(x) - x^2 \left(f(b) - f(a)\right)$$
$$F(a) - F(b) = 0$$

由于 $F(a) = F(b), F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x) = \left(b^2-a^2\right) \frac{d}{dx} f(x) - 2x\left(f(b)-f(a)\right)$$

据此可得:

$$\left(b^2-a^2\right) \frac{d}{dx} f(\xi) = 2\xi\left(f(b)-f(a)\right)$$

命题得证.

例 16: 若 $f(x), g(x)$ 在 $[a,b]$ 连续, (a,b) 可导, 则存在 $\xi \in (a,b)$, 使

$$f(a)g'(\xi) - g(a)f'(\xi) = \frac{g(b)f(a)-f(b)g(a)}{(b-a)} \tag{A.6}$$

```
> myproc((f(a)*g(b)-f(b)*g(a))/(b-a)=f(a)*diff(g(x),x)-g(a)*diff(f(x),x), x);
```

辅助函数为:

$$F(x) = \frac{\left(f(a)g(b)-f(b)g(a)\right)x}{b-a} - f(a)g(x) + g(a)f(x)$$
$$F(a) - F(b) = 0$$

由于 $F(a) = F(b), F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x) = \frac{g(b)f(a)-f(b)g(a)}{b-a} - f(a)\frac{d}{dx}g(x) + g(a)\frac{d}{dx}f(x)$$

据此可得:

$$\frac{g(b)f(a)-f(b)g(a)}{b-a} = f(a)\frac{d}{dx}g(\xi) - g(a)\frac{d}{dx}f(\xi)$$

命题得证.

例 17: 若 $f(x), g(x)$ 在 $[a,b]$ 连续, (a,b) 可导, 则存在 $\xi \in (a,b)$, 使

$$\frac{bf(b)-af(a)}{b-a} = f(\xi) + \xi f'(\xi) \tag{A.7}$$

```
> myproc((b*f(b)-a*f(a))/(b-a)=f(x)+x*diff(f(x),x), x);
```

辅助函数为:

$$F(x) = \frac{(bf(b)-af(a))\,x}{b-a} - xf(x)$$

$$F(a) - F(b) = 0$$

由于 $F(a) = F(b)$, $F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x) = \frac{bf(b)-af(a)}{b-a} - f(x) - x\frac{d}{dx}f(x)$$

据此可得:

$$\frac{bf(b)-af(a)}{b-a} = f(x) + \xi\frac{d}{dx}f(\xi)$$

命题得证.

例 18: 若 $f(x), g(x)$ 在 $[a,b]$ 连续, (a,b) 可导, 且 $g'(x) \neq 0$, 则存在 $\xi \in (a,b)$, 使

$$\frac{f'(\xi)}{g'(\xi)} = \frac{f(a)-f(\xi)}{g(\xi)-g(b)} \tag{A.8}$$

```
> myproc(diff(f(x),x)/diff(g(x),x)=(f(a)-f(x))/(g(x)-g(b)), x);
```

辅助函数为:

$$F(x) = -g(b)f(x) - f(a)g(x) + f(x)g(x)$$

$$F(a) - F(b) = 0$$

由于 $F(a) = F(b)$, $F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x) = \left(\frac{d}{dx}f(x)\right)g(x) + f(x)\frac{d}{dx}g(x) - \left(\frac{d}{dx}f(x)\right)g(b) - f(a)\frac{d}{dx}g(x)$$

据此可得:

$$\frac{\frac{d}{dx}f(x)}{\frac{d}{dx}g(x)} = \frac{f(a)-f(x)}{g(x)-g(b)}$$

命题得证.

例 19: 若 $f(x), g(x), h(x)$ 在 $[a,b]$ 连续, (a,b) 可导, 且 $g'(x) \neq 0$, 则存在 $\xi \in (a,b)$, 使

$$f(b)g(b)h(b)-f(a)g(a)h(a)=(f'(\xi)g(\xi)h(\xi)+f(\xi)g'(\xi))h(\xi)+f(\xi)g(\xi)h'(\xi))(b-a) \tag{A.9}$$

```
> myproc(f(b)*g(b)*h(b)-f(a)*g(a)*h(a)= (diff(f(x), x)*g(x)*h(x)+f(x)*diff(g(x), x)
    *h(x)+f(x)*g(x)*diff(h(x),x))*(b-a), x);
```

辅助函数为:

$$F(x)=f(b)g(b)h(b)x-f(a)g(a)h(a)x+f(x)g(x)h(x)a-f(x)g(x)h(x)b$$

$$F(a)-F(b)=0$$

由于 $F(a)=F(b), F(x)$ 满足 Rolle 定理条件, 于是有:

$$\begin{aligned}F'(x)=&f(b)g(b)h(b)-f(a)g(a)h(a)-(\tfrac{d}{dx}f(x))g(x)h(x)b+(\tfrac{d}{dx}f(x))g(x)h(x)a\\&-f(x)(\tfrac{d}{dx}g(x))h(x)b+f(x)(\tfrac{d}{dx}g(x))h(x)a-f(x)g(x)(\tfrac{d}{dx}h(x))b\\&+f(x)g(x)(\tfrac{d}{dx}h(x))a\end{aligned}$$

据此可得:

$$f(b)g(b)h(b)-f(a)g(a)h(a)=(f'(\xi)g(\xi)h(\xi)+f(\xi)g'(\xi))h(\xi)+f(\xi)g(\xi)h'(\xi))(b-a)$$

命题得证.

例 20: 若 $f_i(x)(i=1,2,\ldots,n)$ 在 $[a,b]$ 连续, (a,b) 可导, 且

$$\sum_{i=1}^{n}\lambda_i=0$$

则存在 $\xi \in (a,b)$, 使

$$\sum_{i=1}^{n}\lambda_i\frac{f_i'(\xi)}{f_i(b)-f_i(a)}=0 \tag{A.10}$$

这是 Cauchy 中值定理的推广形式. 事实上, 当 $n=2$ 时, 该命题即为 Cauchy 中值定理. 这里, 用我们的程序 *myproc* 推证 $n=4$ 时的情形, 其余类推. 但应用该程序时需要增加一个假设条件 $\sum_{i=1}^{n}\lambda_i=0$, 用`Maple`中的命令`assume`即可.

```
> assume(lambda[1]+lambda[2]+lambda[3]+lambda[4]=0);
> myproc(lambda[1]*diff(f[1](x),x)/(f[1](b)-f[1](a))+lambda[2]*diff(f[2](x), x)/(f[2](b)
```

```
-f[2](a))+lambda[3]*diff(f[3](x), x)/(f[3](b)-f[3](a))+lambda[4]*diff(f[4](x), x)
/(f[4](b)-f[4](a))=0, x);
```

辅助函数为:

$$F(x)=\frac{\lambda_1^{\sim} f_1(x)}{f_1(b)-f_1(a)}+\frac{\lambda_2^{\sim} f_2(x)}{f_2(b)-f_2(a)}+\frac{\lambda_3^{\sim} f_3(x)}{f_3(b)-f_3(a)}+\frac{\lambda_4^{\sim} f_4(x)}{f_4(b)-f_4(a)}$$

$$F(a)-F(b)=0$$

由于 $F(a)=F(b), F(x)$ 满足 Rolle 定理条件, 于是有:

$$F'(x)=\frac{\lambda_1^{\sim} \frac{d}{dx} f_1(x)}{f_1(b)-f_1(a)}+\frac{\lambda_2^{\sim} \frac{d}{dx} f_2(x)}{f_2(b)-f_2(a)}+\frac{\lambda_3^{\sim} \frac{d}{dx} f_3(x)}{f_3(b)-f_3(a)}+\frac{\lambda_4^{\sim} \frac{d}{dx} f_4(x)}{f_4(b)-f_4(a)}$$

据此可得:

$$\frac{\lambda_1^{\sim} \frac{d}{dx} f_1(x)}{f_1(b)-f_1(a)}+\frac{\lambda_2^{\sim} \frac{d}{dx} f_2(x)}{f_2(b)-f_2(a)}+\frac{\lambda_3^{\sim} \frac{d}{dx} f_3(x)}{f_3(b)-f_3(a)}+\frac{\lambda_4^{\sim} \frac{d}{dx} f_4(x)}{f_4(b)-f_4(a)}=0$$

命题得证.

A.4 本章小结

符号积分问题是微积分及计算机代数系统中的难点. 本章运用数学分析及计算机代数基本原理和方法解决了Maple及Matlab下符号积分缺陷, 并将其应用于微分中值定理自动推证中. 结果表明: 所建立的符号积分算法思路简单, 操作性强, 相应的计算机程序功能强大, 具有一定的推广价值.

但是, 由于符号积分本身的困难, 在微分中值定理自动推证过程中, 利用程序得到的辅助函数也并非是最简单的形式, 甚至有些题目 (如含有不等式等) 利用此程序尚无法实现自动推证, 因此, 该算法有待进一步完善.

符号积分无疑是微分方程求解的基础之一, 这将是符号积分后续研究的一个问题.

另外, 需要说明的一个问题是, 计算机代数系统Maple自 Version 11 始 (2005 年 7 月发布), 对原有计算引擎进行了大规模的改进, 其中包括对积分函数int的完善, 完善后的int的符号积分功能较我们 2003 年建立的myint算法有过之而无不及.

参考文献①

[1] J. Anttila, A spline collocation method for parabolic pseudodifferential equations Journal of Computational and Applied Mathematics. 140:41-61,2002.

[2] K. Atkingson, A survey of numerical methods for solving nonlinear integral equations, Journal Of Integral Equations and Applications. 4: 15-46, 1992.

[3] E. Babolian, A. Davari, Numerical implementation of Adomian decomposition method for linear Volterra integral equations of the second kind, Applied Mathematics and Computation. 165: 223-227, 2005

[4] B. A. Beltyukov, L. N. Kuznechikhina, A Runge-Kutta method for the solution of two-dimensional nonlinear Volterra integral equations, Differential Equations. 12: 1169-1173, 1976.

[5] J. Biazar, E. Babolian, and R.Islam. Solution of a system of volterra integral equations of the first kind by adomian method. Applied Mathematics and Computation. 139:249-258, 2003.

[6] J. G. Blom, H. Brunner, The numerical solution of nonlinearVolterra integral equations of the second kind by collocation and iterated collocation methods, SIAM Journal of Scientific and Statistical Computing. 8:806-830, 1987.

[7] H. Brunner. Iterated collocation methods and their discretizations for volterra integral equations. SIAM Journal of Numerical Analysis. 21:1132-1145, 1984.

[8] H. Brunner, Y. Yan. On global superconvergence of iterated collocation solutions to linear second-kind volterra integral equations. J.Comp. Appl. Math, 67:187 + 189, 1996.

[9] H. Brunner, A. Makroglou, R. K. Miller. Mixed interpolation collocation methods for first and second Volterra integro-differential equations with periodic solution. Appllied Numerical Mathematics. 23:381-402, 1997.

[10] H. Brunner, A. Makroglou, R. K. Miller. On Mixed collocation methods for Volterra integral equations with periodic solution. Appllied Numerical Mathematics. 24:115-130, 1997.

[11] H. Brunner, Collocation Methods for Volterra Integral and Related Functional Equations, Cambridge University Press, Cambridge, 2004.

[12] G. Capobianco, D. Conte, An efficient and fast parallel method for Volterra integral equations of Abel type, Journal of Computational and Applied Mathematics. 189:481-493, 1999.

[13] A. Cardone, E. Messina, E. Russo, A fast iterative method for discretized Volterra-Fredholm integral equations, Journal of Computational and Applied Mathematics. 189:568-579,2006.

[14] Y. Cao, M. Huang, L. Liu, Y. Xue, Hybrid collocation methods for Fredholm integral equations with weakly singular kernels, Applied Numerical Mathematics. 57:549-561, 2007.

[15] 陈传璋, 候宗仪, 李明忠, 积分方程理论及其应用, 上海: 上海科学技术出版社, 1987

[16] F. Chen, J. Deng, Y. Feng, Algebraic surface blending using Wu's method. Proc of ASCM'00, World Scientific, Singapore, 171-181, 2000.

[17] 陈怀堂, 孤立子与微分几何中某些问题的机械化方法, 博士学位论文, 大连理工大学, 2004.

[18] Y. Chen, A general bijective algorithm for increasing tress, Systems Science and Mathematical Sciences, 12:193-203, 1999.

[19] 陈勇, 孤立子理论的若干问题的研究及机械化实现, 博士学位论文, 大连理工大学, 2003.

[20] S. Chou, X. Gao, Automated Reasoning in Differential Geometry and Mechanics: Part I. An Improved Version of Ritt–Wu's Decomposition Algorithm, Journal of Automated Reasoning, 10: 161-172, 1993.

[21] S. Chou, X. Gao, Automated Reasoning in Differential Geometry and Mechanics: Part II. Mechanical Theorem Proving, Journal of Automated Reasoning, 10: 173-189, 1993.

[22] S. Chou, X. Gao, Automated Reasoning in Differential Geometry and Mechanics: Part III, Bertrand Curves, Journal of System and Mathematics, 6:186-192, 1993.

[23] S. Chou, X. Gao, J. Zhang, Automated Generation of of Readable Proofs with Geometric Invariants I. Multiple and Shortest Proof Generation, Journal of Automated Reasoning, 17:325-347, 1996.

[24] S. Chou, X. Gao, J. Zhang, Automated Generation of of Readable Proofs with Geometric Invariants II. Proving Theorems with Full-Angles, Journal of Automated Reasoning, 17:349-370, 1996.

[25] D. Contea, I. Preteb, Fast collocation methods for Volterra integral equations of convolution type, Journal of Computational and Applied Mathematics. 196:652-663, 2006.

[26] L. M. Delves, J. L. Mohamed, Computational methods for integral equations, Cambridge University Press, Cambridge, 1985

[27] V. Dominguez, High-order collocation and quadrature methods for some logarithmic kernel integral equations on open arcs, Journal of Computational and Applied Mathematics. 161:145-159, 2003.

[28] M. Enculescu, M. Bestehorn, Mechanisms for activity spread in a neural field model Neurocomputing. 70:2996-3000, 2007.

[29] 范恩贵, 可积系统与计算机代数. 北京: 科学出版社,2004.

[30] 范恩贵. 孤立子和可积系统. 博士学位论文, 大连理工大学, 1998.

[31] 范恩贵, 张鸿庆. 非线性孤子方程的齐次平衡法. 物理学报, 47:353-362, 1998.

[32] I. Fredholm, Surune nonvelle methode pour la resolution du problem de Dirichlet, Overs Vet-Ak, Stockholm, 1900.

[33] I. Fredholm, Surune classe d'equation fonctionnelle, Acta Mathematica, 27, 1993.

[34] 符红光, 杨路, 曾振柄, 构造广义 Sturm 序列的递归算法, 中国科学 (E), 29:546-555, 1999.

[35] X. Gao, S. Chou, A Zero Structure Theorem for Differential Parametric Systems, Journal of Symbolic Computation, 16:585-595, 1993.

[36] X. Gao, S. Chou, Solving Geometric Constraint Systems I. A Global Propagation Approach, Computer Aideded Design, 30:47-54, 1998.

[37] X. Gao, S. C. Chou, Solving Geometric Constraint Systems II. A Symbolic Computational Approach, Computer Aided Design, 30:115-122,1998.

[38] X. Gao, J. Zhang, S. Chou, Geometry Expert(in Chinese), Nine Chapters Pub., Taiwan, 1998.

[39] 高小山, 数学机械化进展综述, 数学进展, 30:385-404, 2001.

[40] 高小山, 王定康, 裘宗燕, 杨宏, 方程求解与机器证明, 北京: 科学出版社,2006.

[41] H. S. Goghary, S. Javadi, E. Babolian, Restarted Adomian method for system of nonlinear Volterra equations, Appl.Math.Comp., 161(2005)745-751

[42] A. Golbabai, S. Seifollahi, Numerical solution of the second kind integral equations using radial basis function networks, Applied Mathematics and Computation. 174:877-883, 2006.

[43] 龚文发, 沈有建, 李芳, 解第二类非线性 Fredholm 积分方程的小波 Galerkin 方法, 海南师范学院学报 (自然科学版), 20:105-110, 2007.

[44] 郭大钧, 孙经先, 非线性积分方程, 济南: 山东科学技术出版社, 1989.

[45] G. Han, H. Ken, S. Kokichi, J. Wang, Extrapolation method of iterated collocation solution for two-dimensional nonlinear Volterra integral equations, Applied Mathematics and Computation. 112:49-61, 2000.

[46] G. Han, R. Wang, Richardson extrapolation of iterated discrete Galerkin solution for two-dimensional Fredholm integral equations, Journal of Computational and Applied Mathematics. 139:49-63, 2002.

[47] 何力军, 吕国志, 求解桥联积分方程的一种新的数值方法, 西北工业大学学报, 20:241-243, 2002.

[48] D. Hilbert, Grundzüge either allgemeinen Theorie der linearen Integralgleichungen, Chelsea, 1912 (reprint in 1953).

[49] S. M. Hosseini, S. Shahmorad, Tau numerical solution of Fredholm integro-differential equations with arbitrary polynomial bases Applied Mathematical Modelling, 27:145-154, 2003.

[50] V. Horvat, M. Rogina, Tension spline collocation methods for singularly perturbed Volterra integro-differential and Volterra integral equations, Journal of Computational and Applied Mathematics. 140:381-402,2002.

[51] J. Huang, T. Lv. The mechanical quadrature methods and their splitting extrapolations for solving first-kind boundary integral equations on polygonal regions. Mathematical Numerical Sinica. 1:51-60, 2004.

[52] 黄秋梅, 杨一都, Fredholm 积分方程特征值问题配置法外推的 Matlab 实验, 数学的实践与认识, 37:163-168, 2007.

[53] J. E. Hurtado, Analysis of one-dimensional stochastic finite elements using neural networks Probabilistic Engineering Mechanics. 17:35-44,2002.

[54] 基斯 · 德夫林 (李文林, 袁向东, 李家宝, 包芳勋译), 数学: 新的黄金时代, 上海: 上海教育出版社, 1997.

[55] 纪志刚, 吴文俊与数学机械化, 上海交通大学学报 (社科版), 9: 13-18, 2001.

[56] 姜海波, 杜金元, 具一阶奇异性解的奇异积分方程的直接解法, 数学杂志, 27:222-226, 2007.

[57] P. Junghanns, K. Muller, A collocation method for nonlinear Cauchy singular integral equations, Journal of Computational and Applied Mathematics. 115:283-300, 2000.

[58] R. Kangro, P. Oja, Convergence of spline collocation for Volterra integral equations Applied Numerical Mathematics. doi:10.1016/j.apnum.2007.08.004, 2007.

[59] D. Kapur, J. Mundy, Wu's method and its application to prespective viewing. Artificial Intelligence, 37:15-36, 1988.

[60] M. Kline, Mathematical thought from ancient to modern times, Oxfrod University Press, New York, 1972.

[61] P. K. Kythe, P. Puri, Computational methods for linear integral equations, Birkhauser Bosten, c/o Springer-Verlag, New York, Inc., 175 Fifth Avenue, New York, USA, 2002.

[62] 李彪, 孤立子理论中若干精确求解方法的研究及应用, 博士学位论文, 大连理工大学, 2004.

[63] 李大潜, 谈谈现代应用数学, 中国科学院院刊, 18:219-222, 2003.

[64] 李德生, 若干非线性演化方程精确求解法的研究, 博士学位论文, 大连理工大学, 2004.

[65] H. Li, Clifford algebra approaches to mechanical geometry theorem proving. In: Mathematics Mechanication (X. Gao, D. Wang Eds.), Academic Press, London, 205-229, 2000.

[66] 李文林, 数学史教程, 北京: 高等教育出版社, 2000, 176-177.

[67] 李信富, 李小凡, 张美根, 地震波数值模拟方法研究综述, 防灾减灾工程学报, 27:241-248, 2007.

[68] Z. Li, M. Singer, M. Wu, D. Zheng, Determining the one-dimensional submodules of Laurent-Ore modules. Proc of ISSAC'2006, 220-272, ACM Press, 2006.

[69] Z. Li, M. Wang, Travelling wave solutions to the two-dimensional KdV-Burgers equation. Journal of Physics A: Mathematics And General, 26:6027-6031, 1993.

[70] Z. Li, H. Shi, Exact solutions for Belousove-Zhabotinskii reaction diffusion system, Applied Mathematics-Journal of Chinese University, 11B: 1-6, 1996.

[71] Z. Li, Y. Liu, Rath: A Maple package for finding travelling solitary wave solutions to nonlinear evolution equations. Computer Physics Communications, 148:256-266, 2002.

[72] Z. Li, Y. Liu, RAEEM: A Maple package for finding a series of exact traveling wave solutions for nonlinear evolution equations Computer Physics Communications, 163: 191-201, 2004.

[73] 李志斌, 非线性数学物理方程的行波解, 北京: 科学出版社, 2007.

[74] X. Liang, M. Liu, X. Che, Solving second kind integral equations by Galerkin methods with continuous orthogonal wavelets, Journal of Computational and Applied Mathematics. 136:149-161, 2001.

[75] P. Lima, T. Diogo. Numerical solution of a nonuniquely solvable volterra integral equation using extrapolation methods. Journal of Computational and Applied Mathematics. 140:537-557, 2002.

[76] D. Lin, Z. Liu, Some results on theorem proving in geometry over finite fields. In: Proc IWMM'92(W. Wu, M. Chen Eds.), International Academic, Beijing, 222-235, 1992.

[77] 刘唐伟, 应正卫, 吴志强, 第二类非线性 Fredholm 型积分方程数值解, 东华理工学院学报, 28:294-296, 2005.

[78] 刘天佑, 杨宇山, 李媛媛, 大型积分方程降阶解法与重力资料曲面延拓, 20:290-296, 2007.

[79] Y. Liu, Z. Li, An Automated Algebraic Method for finding a series of exact travelling wave solutions of nonlinear evolution equations. Chinise Physics Letter, 20:317-320, 2003.

[80] A. T. Lonseth, Sources and applications of integral eqautions, SIAM Review, 19:241-278, 1977.

[81] G. Long, G. Nelakantib, Iteration methods for Fredholm integral equations of the second kind, Computer Mathematics with Appllication. 53: 886-894, 2007.

[82] 隆广庆, 强奇异积分方程小波 Petrov-Galerkin 快速算法, 中山大学学报 (自然科学版), 45:1-4, 2006.

[83] 陆征一, 微分学基本定理论证的积分辅助函数法, 高等教学研究, 6(4): 29-31, 2003

[84] 陆征一, 何碧, 罗勇, 多项式系统的实根分离算法及其应用. 北京: 科学出版社,2004.

[85] 吕和祥, 于洪洁, 动力学方程的积分型直接积分法, 应用数学和力学, 22:151-156, 2001.

[86] 吕和祥, 于洪洁, 动力学方程的解析逐步积分法, 工程力学, 18:1-7, 2001.

[87] 吕卓生, 计算微分方程对称与精确解的机械化算法及实现, 博士学位论文, 大连理工大学, 2003.

[88] T. Lv, Y. Huang, Extrapolation method for solving weakly singular nonlinear Volterra integral equations of the second kind, Journal of Mathematical Analysis and Applications. 324:225-237, 2006.

[89] H. Ma, Q.H. Qin, Boundary integral equation supported differential quadrature method to solve problems over general irregular geometries, Computer Mech. 36: 21-33, 2005.

[90] K. Maleknejad, M.Shahrezaee. Using Runge-Kutta method for numerical solution of the system of volterra equation. Applied Mathematics and Computation. 149:399-410, 2004.

[91] K. Maleknejad, H. Safdari, Parallel algorithm for solving linear systems arising from PDE and integral equations, Applied Mathematics and Computation. 151:443-453, 2004.

[92] K. Maleknejad, F. Mirzaee, Using rationalized Haar wavelet for solving linear integral equations, Applied Mathematics and Computation. 160:579-587, 2005.

[93] K. Maleknejad, N. Aghazadeh, Numerical solution of Volterra integral equations of the second kind with convolution kernel by using Taylor-series expansion method, Applied Mathematics and Computation. 161:915-922, 2005.

[94] K. Maleknejad, A. Arzhang, Numerical solution of the Fredholm singular integro-differential equation with Cauchy kernel by using Taylor-series expansion and Galerkin method, Applied Mathematics and Computation. 182:888-897, 2006.

[95] K. Maleknejad, H. Derili, Numerical solution of Hammerstein integral equations by using combination of spline-collocation method and Lagrange interpolation, Applied Mathematics and Computation. 190:1557-1562, 2007.

[96] www.maplesoft.com.

[97] 梅建琴, 微分方程组精确解及其解的规模的机械化算法, 博士学位论文, 大连理工大学, 2006.

[98] M. Muhammad, A. Nurmuhammad, M. Moria, M. Sugihara, Numerical solution of integral equations by means of the Sinc collocation method based on the double exponential transformation, Journal of Computational and Applied Mathematics. 177:269-286, 2005.

[99] M. T. Naira, S. V. Pereverzevb, Regularized collocation method for Fredholm integral equations of the first kind, Journal of Complexity, doi:10.1016/j.jco.2006.09.002, 2006.

[100] 宁德志, 滕斌, 张晓兔, 势流问题边界积分方程的几种解法对比, 辽宁工程技术大学学报 (自然科学版), 24:835-837, 2005.

[101] P. Oja, D. Saveljeva. Cubic spline collocation for integral equations. Computing. 69:319-337, 2002.

[102] K. Parand, M. Razzaghi, Rational Chebyshev tau method for solving Volterra's population model Applied Mathematics and Computation. 149:893-900,2004.

[103] A. Pedas, E. Tamme, Discrete Galerkin method for Fredholm integro-differential equations with weakly singular kernels, Journal of Computational and Applied Mathematics. doi:10.1016/j.cam.2006.12.024, 2006.

[104] H. Poincare, Lecons de mecanique celests, Vol.III, Paris, 1910.

[105] J. Pour-Mahmoud, M.Y. Rahimi-Ardabili, S. Shahmorad, Numerical solution of the system of Fredholm integro-differential equations by the Tau method Applied Mathematics and Computation. 168:465-478, 2005.

[106] M. Rabbani, K. Maleknejad, N. Aghazadeh, R. Mollapourasl, Computational projection methods for solving Fredholm integral equation, Applied Mathematics and Computation. 191:140-143, 2007

[107] M. Y. Rahimi-Ardabili, S. Shahmorad, Iterative numerical solution of non-linear integro-differential equations by the Tau method Applied Mathematics and Computation, doi:10.1016/j.amc.2007.04.002, 2007.

[108] M. T. Rashed. Numerical solution of functional differential, integral and integro-differential equations. Applied Mathematics and Computation. 156:485-492, 2004.

[109] J. Rashidinia, M. Zarebnia, Numerical solution of linear integral equations by using Sinc-collocation method, Applied Mathematics and Computation. 168:806-822, 2005.

[110] Y. Ren, B.Zhang, H. Qiao. A simple taylor-series expansion method for a class of second kind integral equations. Journal of Computational and Applied Mathematics. 110:15-24, 1999.

[111] B. Sepehrian, M. Razzaghi, Single-term Walsh series method for the Volterra integro-differential equations, Engineering Analysis with Boundary Elements. 28:1315-1319, 2004.

[112] S. Shahmorad, Numerical solution of the general form linear Fredholm + Volterra integro-differential equations by the Tau method with an error estimation Applied Mathematics and Computation. 167:1418-1429, 2005.

[113] X. Shang, D. Han, Numerical solution of Fredholm integral equations of the first kind by using linear Legendre multi-wavelets, Applied Mathematics and Computation. 191:440-444, 2007.

[114] 沈以淡, 积分方程, 北京: 北京理工大学出版社, 1989.

[115] H. Shi, Solving of the Yang-Baxter equation, Proc. of IWMM92, International Academic Publication, Beijing, 1992.

[116] H. Shi, Chern classes of surface with singularities, Proc. of Inter Seminar on Singularity and Complex Geometry, International Press, Beijing, 1994.

[117] 石赫, 数学机械化引论, 长沙: 湖南教育出版社, 1998.

[118] 石永福, 王立群, 现代数学技术及其影响, 西北师范大学学报 (自然科学版), 41:94-97, 2005.

[119] B. G. Sloss, W. F. Blyth, A Walsh function method for a non-linear Volterra integral equation, Journal of the Franklin Institute. 340:25-41, 2003.

[120] B. Xia, L. Yang, An algorithm for isolating the real solutions of semi- algebraic systems and its applications, Journal of Symbolic Computation, 34:461-477, 2002.

[121] 夏明耀, 伍振兴, 基于单积分方程矩量法的海洋表面微波散射模拟, 电子学报, 33:385-388, 2005.

[122] 夏铁成, 吴方法及其在偏微分方程中的应用, 博士学位论文, 大连理工大学, 2002.

[123] X. Tang, Z. Pang, T. Zhu, J. Liu, Wavelet Numerical Solutions for Weakly Singular Fredholm Integral Equations of the Second Kind, 武汉大学学报 (自然科学英文版), 12:437-441, 2007.

[124] G. Vainikko, A. Kivinukk, J. Lippus, Fast solvers of integral equations of the second kind: wavelet methods, Journal of Complexity. 21:243-273, 2005.

[125] V. Vemuri, G. Jang, Inversion of Fredholm integral equations of the first kind with fully connected neural networks,Journal of the Franklin Institute. 329:241-257, 1992.

[126] 王德明, 稳定求解第一类 Fredholm 积分方程的一个方法, 同济大学学报 (自然科学版), 34:1414-1416, 2006.

[127] D. Wang, A New Theorem Discovered by Computer Prover. Journal of Geometry, 36:173-182, 1989.

[128] D. Wang, Mechanical Manipulation for a Class of Differential Systems. Journal of Symbolic Computation. 12:233-254, 1991.

[129] D. Wang, Irreducible Decomposition of Algebraic Varieties via Characteristic Sets and Groebner Bases. Computer Aided Geometric Design 9:471-484, 1992.

[130] D. Wang, A Method for Proving Theorems in Differential Geometry and Mechanics. Journal of Universal Computer Science. 1:658-673, 1995.

[131] D. Wang, Elimination Procedures for Mechanical Theorem Proving in Geometry. Annals of Mathematics and Artificial Intelligence. 13:1-24, 1995.

[132] D. Wang, An Elimination Method for Differential Polynomial Systems I. Systems Science and Mathematical Sciences. 9:216-228, 1996.

[133] D. Wang, Clifford Algebraic Calculus for Geometric Reasoning with Application to Computer Vision. In: Automated Deduction in Geometry (D. Wang, ed.), LNAI 1360, Springer-Verlag, Berlin Heidelberg, 115-140, 1997.

[134] D. Wang, S. Fevre, Combining Clifford Algebraic Computing and Term-Rewriting for Geometric Theorem Proving, Fundamenta Informaticae. 39:85-104, 1999.

[135] D. Wang, Elimination Practice: Software Tools and Applications. Imperial College Press, London, 2004.

[136] 王东明, 消去法及其应用, 北京: 科学出版社, 2002.

[137] 王东明, 符号计算选讲, 北京: 清华大学出版社, 2003.

[138] 王东明, 夏壁灿, 计算机代数, 北京: 清华大学出版社, 2004.

[139] 王景, 周后型, 洪伟, 张晓亮, 位积分方程组矩量法用于精确计算非均匀介质柱的电磁散射, 微波学报, 22:1-5, 2006.

[140] 王小林, 一类奇异积分方程组的样条间接近似解法, 数学杂志, 17:113-116, 1997.

[141] 王玮明, 计算机代数系统与符号计算, 兰州: 甘肃科学技术出版社, 2004.

[142] 王玮明, 微分中值定理自动推证研究, 兰州大学学报 (自然科学版), 41(1)99-102, 2005.

[143] W. Wang, X. Lian, A new algorithm for symbolic integral with application, Applied Mathematics and Computation, 162, 949-968, 2005.

[144] W. Wang, A new mechanical algorithm for solving the second kind of Fredholm integral equation, Applied Mathematics and Computation, 172(2)946-962, 2006.

[145] W. Wang, Y. Lin, Z. Zeng, A New Mechanical Algorithm for Solving System of Fredholm Integral Equation Using Resolvent Method, D.-S. Huang et al. (Eds.): ICIC 2008, LNCS 5226, 744-754, 2008.

[146] W. Wang, Mechanical algorithm for solving the second kind of Volterra integral equation, Applied Mathematics and Computation, 173(2)1149-1162, 2006.

[147] W. Wang, A mechanical algorithm for solving the Volterra integral equation, Applied Mathematics and Computation, 172(2)1323-1341, 2006.

[148] W. Wang, Z. Zeng, L. Yang, A new mechanical algorithm for solving system of Volterra integral equation, IEEE: 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, 3:67-72, 2007.

[149] W. Wang, An algorithm for solving the high-order nonlinear Volterra- Fredholm integro-differential equation with mechanization, Applied Mathematics and Computation, 172(1)1-23, 2006.

[150] W. Wang, Z. Li, A mechanical algorithm for solving ordinary differential equation, Applied Mathematics and Computation, 172(3)568-583, 2006.

[151] A. M. Wazwaz, S.A. Khuri, Two methods for solving integral equations, Applied mathematics and computation. 77: 79-89, 1996.

[152] A. M. Wazwaz, A new algorithm for calculating Adomian polynomials for nonlinear operators, Applied mathematics and computation. 3:53-69,2000.

[153] J. Wu, Z. Liu, Well-behaved inference rules for first-order theorem proving, Journal of Automated Reasoning. 21:163-172, 1998.

[154] 吴静, 第二类 Fredholm 积分方程的快速数值解法, 广东轻工职业技术学院学报, 3:1-5, 2004.

[155] 吴静, 林福荣, 某第二类 Fredholm 积分方程的一种数值解法, 汕头大学学报 (自然科学版), 18:11-18, 2003.

[156] W. Wu, On the decision problem and mechanization of theorem proving in elementary geometry. Scientia Sinica. 21:159-172, 1978.

[157] W. Wu, Mechanical theorem proving of differential geometry, Scientia Sinica, 94-102, 1979.

[158] W. Wu, Basic principles of mechanical theorem proving in geometries, Vol.I: Part of elementary geometries. Beijing: Science Press, 1984(in Chinese); Berlin: Springer, 1995(English version).

[159] W. Wu, A constructive theory of differential algebraic geometry. In: Proc. of DD6-Symposium, Shanghai, 497-528, 1984.

[160] W. Wu, Basic principles of mechanical theorem-proving in geometries, System Science and Mathematical Science, 4:207-235, 1984; Re-published in Journal of Automated Beasoning, 2:221-252, 1986.

[161] W. Wu, A mechanization method of geometry and its application I. distances, areas and volumes, Journal of Systems Science and Mathematical Science, 6: 204-216, 1986.

[162] W. Wu, A mechanization method of geometry and its application II. curve paris of Bertrand type, Kexue Tongbao, 17: 1281-1284, 1986.

[163] W. Wu, Mechanical derivation of Newton's Gravitational Laws from Kepler's Laws, MM-Res. Preprints, 1:53-61, 1987.

[164] W. Wu, A mechanization method of geometry and its application III. mechanical proving of polynomial inequalities and equation-solving, System Science and Mathematical Science, 1: 1-17, 1988.

[165] W. Wu, A mechanization method of geometry and its applications IV. Some theorems in planar kinematics, System Science and Mathematical Science, 2:97-109, 1989.

[166] W. Wu, T. Wu, A mechanization method of geometries and its applications V. Solving transcendental equations by algebraic methods, MM-Res. Preprints. 3:30-32, 1989.

[167] W. WU, A mechanization method of geometry and its applications VI. Solving inverse kinematic equations of PUMA-type robots, MM-Res.Preprints. 4:49-54, 1989.

[168] W. Wu, On the foundation of algebraic differential geometryg, System Science and Mathematical Science, 2: 289-312, 1989.

[169] W. Wu, On a projection theorem of quasi-varieties in elimination theory, Chinese Annals of Mathematics, 11B: 220-226, 1990.

[170] W. Wu, On the chemical equlibrium problem and equation-solving, Acta Mathematica Sinica, 10:361-374, 1990.

[171] W. Wu, Automation of Theorem-Proving, Workshop on Advanced Computers for Artificial Intelligence, Beijing, 11-15, 1990.

[172] W. Wu, Mechanical theorem proving of differential geometries and some of its applications in mechanics, Journal of Automated Beasoning, 7:171-191, 1991

[173] W. Wu, A report on mechnical geometry theorem proving, Progress in Natural Science, 2:1-17, 1992.

[174] W. Wu, Equations-solving and theorem-proving: Zero-set formulation and ideal formulation, Proceedings of the Asian Mathematical Conference, 1-10, 1992.

[175] W. Wu, Char-set method of polynomial equation-solving and its applications, Proc. Asian Symposium on Computer Mathematics, 11-21, 1995.

[176] W. Wu, Mathematics Mechanization, Science Press/Kluwer Pub., 2000.

[177] W. Wu, On a Finite Kernel Theorem for Polynomial-Type Optimization Problems and some of its Applications, ISSAC'05, 4, 2005.

[178] 吴文俊, 复兴构造性的数学, 数学进展, 14:334-339, 1985.

[179] J. Xiao, L. Wen, D. Zhang, Solving second kind Fredholm integral equations by periodic wavelet Galerkin method, Applied Mathematics and Computation. 175:508-518, 2006.

[180] 谢福鼎, Wu-Ritt 消元法在偏微分代数方程中的应用, 博士学位论文, 大连理工大学, 2002.

[181] C. Xu, Q. Shi, M. Cheng, A stero vision method based on wavelet and Wu's method. Proc of SBT'95, Beijing, 125-130, 1995.

[182] 徐桂琼, 非线性演化方程的精确解与可积性及其符号计算研究, 博士学位论文, 华东师范大学, 2004.

[183] 徐利明, 聂在平, 胡俊, 半空间跨界面目标电磁散射的精确建模与高效计算, 电子学报, 33:389-392, 2005.

[184] S. Yalcinbas, M.Sezer. The approximate solution of highorder linear Volterra-Fredholm integro-differential equations in terms of Taylor polynomials. Applied Mathematics and Computation. 112:291-308, 2000.

[185] S. Yalcinbas, M. Sezer. Taylor polynomial solutions of nonlinear Volterra-Fredholm integral equations. Applied Mathematics and Computation. 127:195-206, 2002.

[186] Y. Yamazaki, M. Ochiai, A. Holz, T. Hara, Application of neural network algorithm to CAD of magnetic systems, Neurocomputing. 13:217-230,1996.

[187] 闫振亚, 非线性波与可积系统, 博士学位论文, 大连理工大学, 2002.

[188] 杨路, 侯晓荣, 曾振柄, 多项式的完全判别系统, 中国科学 (E), 26:424-441, 1996.

[189] 杨路, 侯晓荣, 夏壁灿, 自动发现不等式型定理的一个完备算法, 中国科学 (E), 31:273-288, 2001.

[190] L. Yang, B. Xia, Automated Deduction in Real Geometry, Geometry Computation, 248-298, World Scientific, 2004.

[191] 杨路, 夏壁灿, 不等式机器证明与自动发现, 北京: 科学出版社, 2008.

[192] 杨路, 夏时洪, 一类构造性几何不等式的机器证明, 计算机学报, 26:769-778, 2003.

[193] L. Yang, J. Zhang, A practical program of automated proving for a class of geometric inequalities, Automated Deduction in Geometry, Lecture Notes in Artificial Intelligence, 41-57, Springer-Verlag, 2001.

[194] L. Yang, J. Zhang, W. Zhang, On number of circles intersected by a line, Journal of Combinatorial Theory(Series A), 98: 395-405, 2002.

[195] 杨德庄, 灵活的数学技术, 数学进展, 34:1-16, 2005.

[196] 杨虎, 郭华东, 基于神经网络方法的极化雷达地表参数反演, 遥感学报, 6:451-455, 2002.

[197] 杨洪勇, 杨昕, 王彦博, A collocation method for the conductivity problem with discontinuous coefficient, 高等学校计算数学学报 (英文版), 14:157-170, 2005.

[198] 杨荣奎, 吕涛, 多角形域上第一类边界积分方程的高精度配置法, 工程数学学报, 21:83-87, 2004.

[199] 姚若侠, 基于符号计算的非线性微分方程精确解及其可积性研究, 博士学位论文, 华东师范大学, 2005.

[200] S. Yousefia, M. Razzaghib, Legendre wavelets method for the nonlinear Volterra-Fredholm integral equations, Mathematics and Computers in Simulation. 70:1-8, 2005.

[201] Y. Yu, L.Wen, S. Li. Nonlinear stability of Runge-Kutta methods for neutral delay integro-differential equations. Applied Mathematics and Computation. 191: 543-549, 2007.

[202] 于亚璇, 非线性方程精确解和一类空间的凸性与光滑性, 博士学位论文, 大连理工大学, 2006.

[203] 云天全, 积分方程及其在力学中的应用, 广州: 华南理工大学出版社, 1990

[204] A. Zerarka, A.Soukeur, A generalized integral quadratic method: I. an efficient solution for one-dimensional volterra integral equation. Communication in Nonlinear Scientific Numerical Simulation. 10:653-663, 2005.

[205] H. Zhang, Applications of mechanical methods to partial differential equations, in Mathe-

matics Mechanization and Applications (X. Gao, D. Wang Eds.), Academic Press, NewYork, 409-432, 2000.

[206] J. Zhang, L. Yang, S Chou, X. Gao, Automated Production of Readable Proofs for Geometry Theorems, Chinese Journal of Computers, 18:380-393, 1995.

[207] J. Zhang, X. Gao, S. Chou, A Geometry Information Search System by Forwad Reasoning, Chinese Journal of Computers, 19:722-727, 1996.

[208] 张景中, 梁松新, 复系数多项式完全判别系统及其自动生成, 中国科学 (E), 29:61-75, 1999.

[209] 张景中, 计算机怎样解几何题, 广州: 暨南大学出版社, 2000.

[210] 张善卿, 微分方程精确解及李对称符号计算研究, 博士学位论文, 华东师范大学, 2004.

[211] S. Zhang, Y.Lin, and M. Rao. Numerical solutions for second-kind Volterra integral equations by Galerkin methods. Appllied Mathematics. 45:19-39, 2000.

[212] 张石生, 积分方程, 重庆: 重庆出版社, 1988.

[213] 张玉峰, 孤立子方程求解与可积系统, 博士学位论文, 大连理工大学, 2002.

[214] 张阳, 薛运华, 求解一类高阶线性 Fredholm 积分微分方程的 Tau 方法, 高等学校计算数学学报, 27:1-5, 2005.

[215] 张志刚, 赵新泉, 利用 BP 人工神经网络计算 Fredholm-II 型积分方程的近似解, 中南民族大学学报, 21: 79-81, 2002.

[216] R. Zhu, J. Huang, T. Lu, Mechanical quadrature methods and their splitting extrapolations for solving boundary integral equations of axisymmetric Laplace mixed boundary value problems, Engineering Analysis with Boundary Elements. 30:390-398, 2006.

[217] 朱佑彬, 傅初黎, 一类不适定问题具备停止规则的简化迭代技巧, 兰州大学学报 (自然科学版), 38:1-6, 2002.

致　谢

在本书即将付梓印刷之际，衷心感谢华东师范大学软件学院曾振柄教授。感谢他在本书的选题、具体问题的研究、算法设计和实现诸方面给予的精心指导。他的指导与帮助，不仅使我在数学机械化和符号计算领域获得了进一步的理解，也使我在人生道路上获得了克服困境的许多教益。

衷心感谢华东师范大学杨路教授、李志斌教授、复旦大学谭永基教授，感谢他们对我学术研究和生活上的指导与帮助。

衷心感谢温州大学数学与信息科学学院院长陆征一教授多年来在学习和工作上给予我的指导与帮助，正是他将我引入符号计算这一研究领域。

衷心感谢中科院植物所李镇清、昆明动物所苏雷、甘肃省农科院院长宋尚有、中北大学靳祯、南京师范大学崔景安、新疆大学腾志东、东北师范大学范猛、哈尔滨工业大学刘胜强、福州大学陈凤德、成都理工大学王文娟、甘肃农业大学杨富民和王兴堂、李允、李双奎等教授以及甘肃省农牧厅闫万贵处长、甘肃科技出版社社长黄培武、韩波及张怀山，感谢他们给予我学术上的指导及生活上的关怀。

衷心感谢华东师范大学软件学院陈勇教授、刘静教授、朱惠彪副教授、蒲戈光副教授、申立勇副教授和赵世忠、吴敏博士给予我的关心与帮助！

衷心感谢我的同事周湘浙、赵焕光、蔡曙光、赵敏、李志国、王德林、胡伟国、王向红、袁建国、樊柄有、吴明江、柳劲松、刘锦明、杨斌、姚峰、励龙昌、陈孟森、周东、蔡联群、陈慧、赵才地、王兴文、丁治民、金文兵、扸延东、李忠月以及戴梅红、林长胜、杨万全、王幼斌、张宗劳、高利新、马万、黎详君、黄玉飞、童帆、龚平娟、郑颖、彭小媚、叶如芳、连新泽、林望等同志给予我的学习和工作上的关心与帮助，也感谢我的同学邹清明、崔建明、周振江、陈良育、单美静、毕忠勤、郭远华、武斌等博士给予我的照顾。感谢我的学生汪海玲、王晓琴、吴宗大、张磊、刁鼎立、饶凤、凌丽、刘厚业、林晔智、杨峰等与我一起学习和讨论！我还要特别感谢首都师范大学美术学院 2008 级硕士生宋烨同学为本书所做的封面设计。

特别感谢甘肃省陇剧院王亨院长、甘肃省教育工委高永新副书记、甘肃农业大学李敏权教授等数十年来给予我生活上的关怀与学习和工作上的帮助。

我还要感谢我的父亲母亲、我的岳父岳母、我的姨父姨姨以及我的兄弟姐妹，他们长期默默的关爱和奉献是我得以安心工作和学习的源动力！

最后，我要特别感谢我的夫人岳延红以及我们可爱的女儿王潇萌，她们的至爱是我生存和奋斗的支撑！这本书也是给她们的.

王玮明

二零零八年九月